PPCPC

JPC

NOVEMBRE

1983

VOLUME-1

NUMERO-8

LE NUMERO 20FF

APPLICATIONS

PHILIPPE CANUEL	5	MATRICES CARREES (3x3) & (4x4)
ROBERT SCHWARTZ	10	PRGM "IF" (suite)
ROBERT SCHWARTZ	15	Fonctions synthétiques et code XROM
JEAN BOSCHAT	17	DROLES DE FONCTIONS
ROBERT SCHWARTZ	21	Fonction WALL
ROBERT SCHWARTZ	21	petites découvertes

PROGRAMMES

JEAN-CLAUDE BECKER	1	COQ / VIN
FREDERIC POUPON	3	SAVEK / GETK
PHILIPPE CANUEL	5	RESOLUTION DE 3 (4) EQUATIONS A 3 (4)INCONUES
ROBERT SCHWARTZ	15	XS & S
ROBERT SCHWARTZ	16	CDX - DCX

LE PETIT THEATRE DES MICROCODES

PHILIPPE GUEZ	13	MOTROM - MNEMONIQUES

LA REVUE DES CODES BARRES

ROBERT SCHWARTZ	19	CB pour stockage de données par séquence
ROBERT SCHWARTZ	20	CB à éxécution directe
ROBERT SCHWARTZ	21	1 Octet de code barre
ROBERT SCHWARTZ	22	Fonctions synthétiques en CB/PRGM
JAKE SCHWARTZ	22	PLANCHES DES INSTRUCTIONS SUR REG. D'ETAT
ROBERT SCHWARTZ	26	FONCTIONS EN CB
ROBERT SCHWARTZ	27	FS
GABRIEL GIL	28	XROM 8,00 à 11,63
GILBERT TISSERAND	32	CB DES FONCTIONS DU XFONCTION
GILBERT TISSERAND	35	CB DES FONCTIONS DU TIME
GILBERT TISSERAND	37	FLAGS EN CODES BARRES
GILBERT TISSERAND	38	E^{10n} ET FIX n EN CODES BARRES

FREDERIC POUPON	3	S.O.S.
JEAN PIERRE SANDOZ	16	S.O.S.
CHOC EN RETOUR	11	VIN6P19 (lepetit théatre)
	12	VIN4 (APPHR & MORSE)
PILOU SANABRIA	2	LE PROF CUISTOT
YVES ALAJOUANINE	10	journées d'études des GEOMETRES-TOPOGRAPHES
	39	BULLETIN D'ABONEMENT AU JOURNAL
	39	BULLETIN D'ADHESION AU PPC P C

IMPRIME ET EDITE PAR LE PPC PARIS CHAPTER 56, RUE JEAN JACQUES ROUSSEAU 75001 PARIS - FRANCE

Le PPC PARIS CHAPTER est une association régie par la loi de 1901 , enregistrée à la préfecture de PARIS le 2 décembre 1982 sous le numero 82/3240.

PRGM:"COQ/VIN"
Nécessaire: Module Horloge
Conseillés: Imprimante
Clavier souple étanche

1450 BYTES
208 REG.
SIZE 003

Début de la préparation A	B	Temps restant C	Entrée Date/Heure du repas D	E
Liste des ingrédients F	G	H	Initialisation I	Relecture du dernier texte J

LBL. utilisés par l'alarme:

"AA" Début de la préparatioh
"CHAM" Préparation des champigons
"PAIN" Préparation du pain de mie
"FIN" Fin de la cuisson

Registres:

000: Heure du repas
001: Date du repas
002: Nombre de convives

OBSERVATIONS:

Ce programme est un guide pour la préparation du coq au vin.
"F" permet de prévoir la quantité de mets à acheter en fonction du nombre de convives.
"D" rappelle au moment voulu qu'il faut se mettre au travail.
"A" débute immédiatement le travail. (durée: 2 H)

Ne pas laisser d'imprimante connectée et sur OFF.

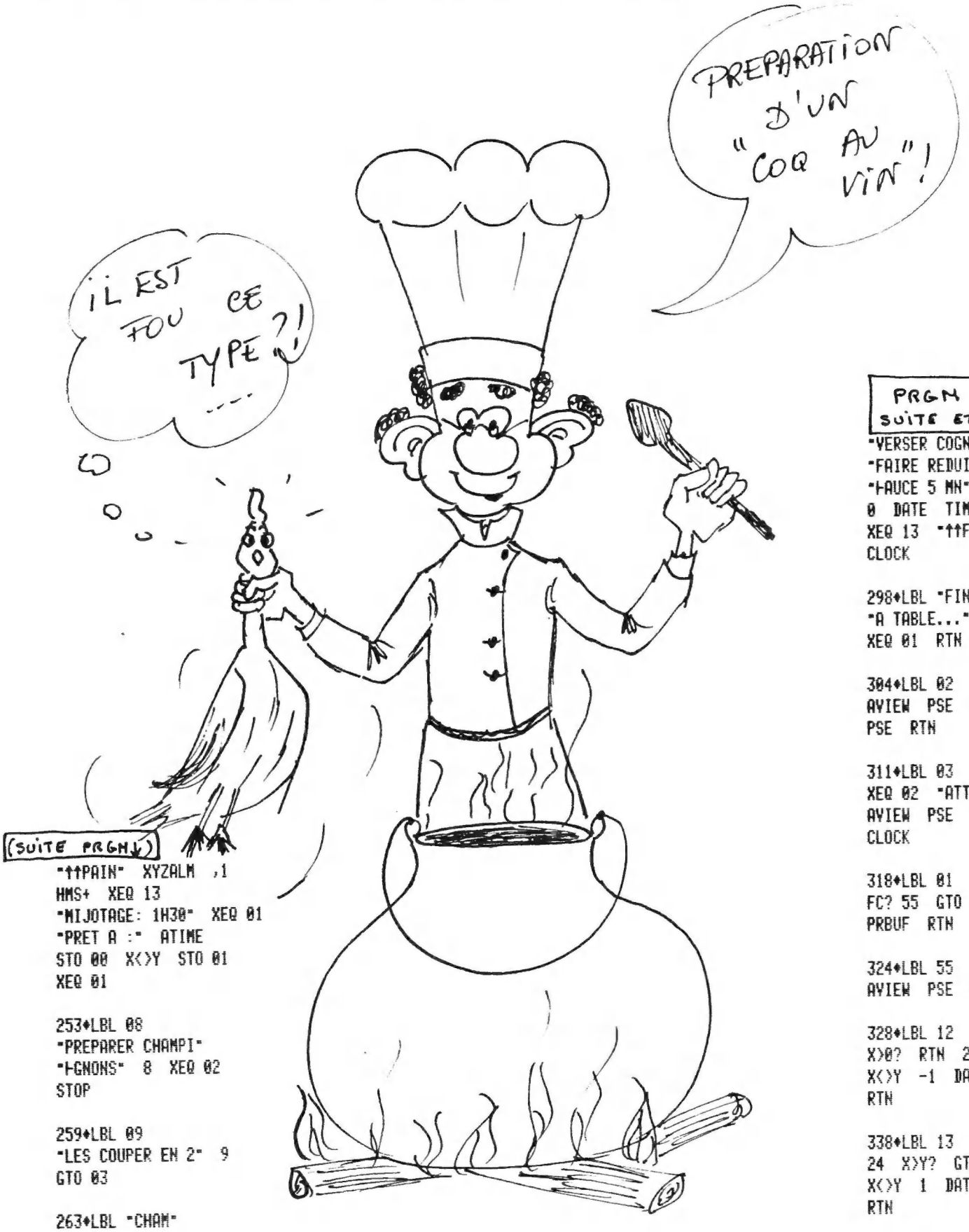
```
LBL'COQ/VIN
LBL'AA
LBL'CHAM
LBL'PAIN
LBL'FIN
.END.        1456 BYTES
                  PRP ""

 01♦LBL "COQ/VIN"
"21/10/1983"

 03♦LBL I
"<A> DEBUT"  AVIEW  PSE
"<C> COMP./REB."  AVIEW
PSE  "<D> PRGM DEBUT"
AVIEW  PSE
"<F> LISTE PRODU"
"⊢ITS"  AVIEW  PSE
"<J> RELECTUR."  AVIEW
PSE  "SIZE 003"  ADV
RTN

 23♦LBL F
CF 29  "Nb. CONVIVES?"
PROMPT  STO 02
"COQ OU POULET:"  AVIEW
PSE  ,333  RCL 02  *
SCI 2  RND  FIX 3  CLA
ARCL X  "⊢ KG"  AVIEW
PSE  "COUPE EN "  RCL 02
2  *  FIX 0  ARCL X
AVIEW  PSE  ,1666
RCL 02  *  SCI 0  RND
FIX 0  1  X<=Y?  X<>Y
CLA  ARCL X
"⊢L. VIN ROUGE"  XEQ 01
"2 CUIL. SOUPE C"
"⊢OGNAC"  XEQ 01  41,5
XEQ 20
"⊢G. CHAMPIGNONS"
XEQ 01  "2 GOUSSES D"
39  XTOA  "AIL ENTIERES"
XEQ 01
"2 GROS OIGNONS "
"⊢EMINCES"  XEQ 01  25
XEQ 20  "⊢G. LARDONS"
XEQ 01  "70"
"⊢G. BEURRE"  XEQ 01
"1 CUIL. SOUPE F"
"⊢ARINE"  XEQ 01
"1 BOUQUET GARNI"  "⊢:"
XEQ 01
"THYM, LAURIER, "
"⊢ROMARIN."  XEQ 01  CLA
FIX 0  ARCL 02
"⊢TRANCHES PAIN"
"⊢ DE MIE"  XEQ 01
"SEL, POIVRE"  XEQ 01
RTN

103♦LBL 20
RCL 02  *  SCI 2  RND
FIX 0  CLA  ARCL X  RTN

112♦LBL C
"PRET a: "  RCL 00
ATIME24  AVIEW  PSE
"DANS: "  TIME  HMS-
DATE  RCL 01  DDAYS  24
*  HMS+  ATIME24  AVIEW
RTN
```

Bonappetitement vôtre,

J-C BECKER

```
130♦LBL J
GTO IND X

132♦LBL D
FIX 2  "DATE REPAS?"
CF 22  PROMPT  FC?C 22
0  0  X<>Y
"HEURE REPAS?"  PROMPT
2  HMS-  XEQ 12  "↑↑AA"
XYZALM  "PREPARER a "
ARCL X  AVIEW  PSE
"LE "  X<>Y  FIX 6
ADATE  AVIEW  BEEP  OFF

159♦LBL "AA"
160♦LBL A
" COQ AU VIN"  SF 12
XEQ 01  CF 12  ADV

166♦LBL 00
"SUR 50 G BEURRE"  "⊢,"
XEQ 01
"COCOTTE FEU VIF"  "⊢,"
XEQ 01
"F. REVEN. MORCE"
"⊢AUX COQ"  0  XEQ 02
STOP

178♦LBL 21
"SANS LE COQ:"  XEQ 01
"OIGNONS, LARDON"  "⊢S"
XEQ 01  "DANS COCOTTE"
21  XEQ 02  STOP

188♦LBL 22
"+ FARINE, MELAN"
"⊢GER,"  22  XEQ 02
STOP

194♦LBL 23
"FAIRE DORER"  23
XEQ 02  STOP

199♦LBL 04
"AJOUTER COQ,"  XEQ 01
"VERSER LE VIN"
"⊢ PEU A PEU"  XEQ 01
"EN TOURNANT"  4  XEQ 02
STOP

209♦LBL 05
"SALER, POIVRER,"  5
XEQ 02  STOP

214♦LBL 06
"BOUQET GARNI + "
"⊢AIL."  6  XEQ 02  STOP

220♦LBL 07
"MELANGER, COUVR"
"⊢IR,"  XEQ 01
"COMMENCER MIJOT"
"⊢AGE"  7  XEQ 02  STOP
0  DATE  TIME  ,15  HMS+
XEQ 13  "↑↑CHAM"  XYZALM
1,05  HMS+  XEQ 13
```

AUJOURD'HUI LE PROF CUISTOT VOUS PRESENTE LA :



(SUITE PRGM↓)

```
"↑↑PAIN"  XYZALM  ,1
HMS+  XEQ 13
"MIJOTAGE: 1H30"  XEQ 01
"PRET A :"  ATIME
STO 00  X<>Y  STO 01
XEQ 01

253♦LBL 08
"PREPARER CHAMPI"
"⊢GNONS"  8  XEQ 02
STOP

259♦LBL 09
"LES COUPER EN 2"  9
GTO 03

263♦LBL "CHAM"
264♦LBL 10
"AJOUTER CHAMPIG"
"⊢NONS"  XEQ 02  10
STOP  "CONTIN. CUISS"
"⊢."  GTO 03

273♦LBL "PAIN"
274♦LBL 11
"FAIRE DORER PAI"
"⊢N MIE"  XEQ 01
"COUPE EN TRIANG"
"⊢LES"  XEQ 02  11  STOP
```

PRGM
SUITE ET FIN (FAIM)↓

```
"VERSER COGNAC"  XEQ 02
"FAIRE REDUIRE S"
"⊢AUCE 5 MN"  AVIEW  PSE
0  DATE  TIME  ,05  HMS+
XEQ 13  "↑↑FIN"  XYZALM
CLOCK

298♦LBL "FIN"
"A TABLE..."  BEEP  BEEP
XEQ 01  RTN

304♦LBL 02
AVIEW  PSE  ADV  TONE 9
PSE  RTN

311♦LBL 03
XEQ 02  "ATTENDEZ..."
AVIEW  PSE  TONE 9
CLOCK

318♦LBL 01
FC? 55  GTO 55  ACA
PRBUF  RTN

324♦LBL 55
AVIEW  PSE  RTN

328♦LBL 12
X>0?  RTN  24  HMS+
X<>Y  -1  DATE+  X<>Y
RTN

338♦LBL 13
24  X>Y?  GTO 14  HMS-
X<>Y  1  DATE+  X<>Y
RTN

348♦LBL 14
RDN  .END.
```

LA SAUVEGARDE DES ASSIGNATIONS EN X-FONCTIONS.

Le programme que je vous propose ce mois ci va vous permettre de ne plus etre oblige de toujours avoir votre lecteur de cartes ou (???) de k7 (capitaliste!) avec vous. En effet, ce que seuls ces periph pouvaient jusqu'a maintenant faire, votre X-Fonctions le peut.

Je vous ferai grace de vous expliquer ce qu'est la sauvegarde des assignations, mais neammoins faites attention !

- Evitez imperativement de sauver vos assignations si certaines concernent des pgms; en effet ces ass. sont tout a fait differentes des autres et vous pourriez rencontrer des problemes au moment du rappel. Par contre, n'hesitez pas a sauvegarder vos ass. synthetiques, c'est meme la principale raison d'etre du programme !

- Le programme n'est pas compatible avec la presence d'alarmes (pas encore mais j'ai confiance en vous.)

- Ne faites pas tourner ce programme en premiere position dans la memoire, mettez au moins un 'end' avant.

Passons maintenànt a l'utilisation, les deux parties de ce pgm ont ete redigees de facon a etre utilisables comme des fonctions standard:

- Pour sauver vos ass. courantes, entrez un nom de fichier non encore attribue en Alpha et lancez la fonction 'SAVEK'. Attention, si le pgm s' arrete sur un message d'erreur, il faut absolument vous debrouiller pour pouvoir relancer le programme afin qu'il puisse s'achever (liberez de la place, changez de nom, ...) car vos assignations sont momentanement toutes inutilisables et pourraient vous donner des sueurs froides. Rien a craindre par contre si vous faites ce que je vous conseille (SANS eteindre la machine !!!).
- Pour rappeler des assignations, assurez vous qu'il y a assez de place pour les recevoir (Sinon je ne reponds de strictement rien), c'est a dire la taille du fichier concerne moins 2 regs, entrez le nom du fichier concerne en Alpha, et lancez 'GETK'. Remuez, c'est pret !

Voila ! Quand vous l'aurez utilise 2 fois, vous aurez tout compris.

Allez, salut et a la prochaine !

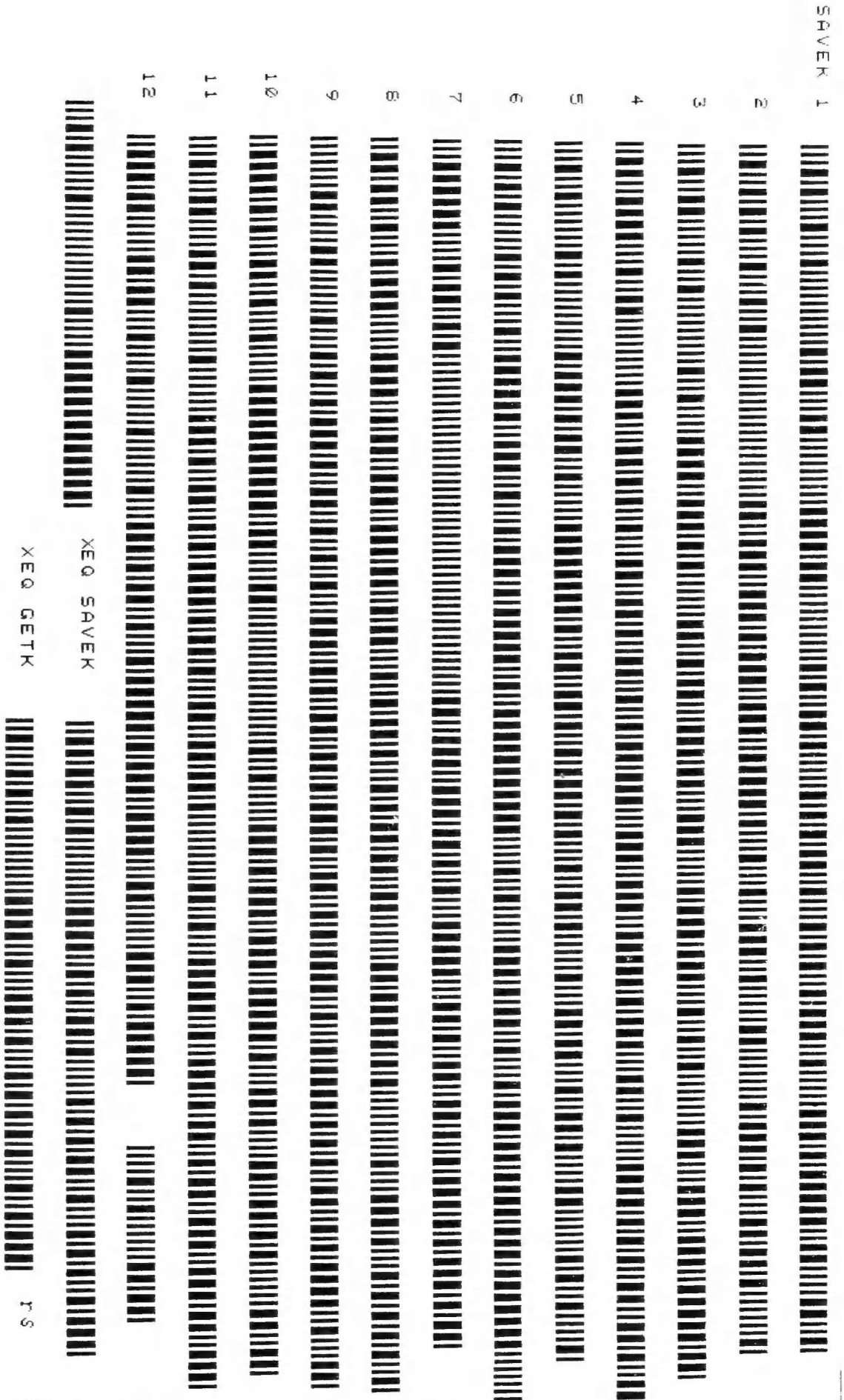
Frederic POUPON (P 35)

S. O. S.

J'en appele a ceux qui savent: a quand un clavier azerty accentue sur HP 75 et 41 ? (P35).

S. O. S.

Qui m'offrira un Drive cassette pour avoir la primeur de mon nouveau compilateur ? Reponse assuree ! (P35).




```
                    PRP ""

 01♦LBL "SAVEK"
RCL [  "µ♦♦"  X<> [
X<> c  1-  ENTER↑

 08♦LBL 00
RDN   E  +  RCL IND X
SF 25  X≠0?  GTO 00  CLX
2  +  X<>Y  X<> c  X<>Y
CF 25  CRFLD  X<>Y
X<> c  X<>Y  RCL '
SAVEX  RDN  RCL e  SAVEX
RDN  3  -   E3  /

 37♦LBL 01
RCL IND X  STO [  "├"
RCL [  STO \  "├♦♦♦♦♦♦"
X<> \  STO IND Z  SAVEX
RDN  RDN  ISG X  GTO 01
GTO 03

 52♦LBL "GETK"
CLKEYS  -  SEEKPTA  GETX
STO '  GETX  STO e
"µ♦♦"  RCL [  X<> c  .9
SF 25

 65♦LBL 02
GETX  FC? 25  GTO 03
STO IND Y  RDN  ISG X
GTO 02

 73♦LBL 03
RDN  STO c  CLST  END
                    CAT 1
LBL'SAVEK
LBL'GETK
END          152 BYTES
```

CODES HEXA

```
   C600F600534156454B90
75F30C0000CE75CE7D111C83
01751B4090F3A81963B1C177
124071CE7D71A919A64B71CE
7D71907AA66975907FA66975
13411B13430290F39175F27F
F090759176F77F0000000000
00CE7691F1A66975759673B2
C4B456CC0DF5004745544BA6
491CA66BA657917AA657917F
F30C00009075CE7D1A19A819
03A657AD19B40191F2759673
B3820475917D73000000CC07
2D87
```

L3: 243,12,0,0,
L40: 242,127,240,
L43: 247,127,0,0,0,0,0,0
,
L60: 243,12,0,0,

Attention, le code de la ligne 54 est 28.
Elle peut etre remplacee par clx.

Réalisation de programmes pour :

1°/ "AA" : calcul du déterminant d'une matrice carrée [3,3]
2°/ "BB" : calcul d'un système de 3 équations à 3 inconnues (x,y,z)
3°/ "CC" : calcul du déterminant d'une matrice carrée [4,4]
4°/ "DD" : calcul d'un système de 4 équations à 4 inconnues (x,y,z,t)

Pour les programmes "BB", "CC", "DD" il y aura souvent besoin de la fonction x< >.. il sera donc intéressant de créer d'abord une fonction personnalisée (USER) : par exemple $\boxed{x \geq y}$ représentant x< >..

1°/ Déterminant d'une matrice carrée [3,3] (programme "AA")

Soit Δ de $[a_{ij}]$ pour $i,j=1,2,3 \longrightarrow \Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$

Les valeurs a_{ij} devront être introduites dans les repères ij. Ex : a_{11} STO 11, ... (9 fois)

	LBL T AA		STOP (a_{23})		*	40	+
	LBL ΦΦ		STO 23		−		Rcl 13
	TONE 4		STOP (a_{31})		Rcl 22		Rcl 32
	STOP (a_{11})		STO 31	30	*		*
	STO 11		STOP (a_{32})		Rcl 12		Rcl 12
	STOP (a_{12})		STO 32		Rcl 31		Rcl 33
	STO 12	20	STOP (a_{33})		*		*
	STOP (a_{13})		STO 33		Rcl 11		−
	STO 13		Rcl 11		Rcl 32		Rcl 21
10	STOP (a_{21})		Rcl 33		*		*
	STO 21		*		−	50	+
	STOP (a_{22})		Rcl 13		Rcl 23		STO ΦΦ
	STO 22		Rcl 31		*		GTO ΦΦ
						(53)	END

Une fois le programme appelé (XEQ α AA α) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, le déterminant Δ peut être connu en faisant Rcl ΦΦ.

De nouvelles valeurs peuvent être dès lors introduites.

2/ Système de 3 équations à 3 inconnues (x, y, z) (programme "BB")

Soit le système suivant :

$$\left\{\begin{array}{l} a_{11}x + a_{12}y + a_{13}z = a_{14} \\ a_{21}x + a_{22}y + a_{23}z = a_{24} \\ a_{31}x + a_{32}y + a_{33}z = a_{34} \end{array}\right\} \equiv \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{[3,3]} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{[3,1]} = \underbrace{\begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}}_{[3,1]}$$

Les valeurs a_{ij} pour $i = 1,2,3$; $j = 1,2,3,4$ devront être introduites dans les repères ij. Ex : a_{11} STO 11, ... (12 fois)

	LBL' BB			STO 34				x<>32			*
	LBL ØØ			XEQ Ø1				STO 34		80	−
	TONE 4			STO ØØ	(i)			XEQ Ø1			Rcl 22
	STOP	(a_{11})	30	Rcl 14				Rcl ØØ			*
	STO 11			x<>11				:			Rcl 12
	STOP	(a_{12})		STO 14				STO Ø2	(y)		Rcl 31
	STO 12			Rcl 24				Rcl 14			*
	STOP	(a_{13})		x<>21		60		x<>12			Rcl 11
	STO 13			STO 24				x<>13			Rcl 32
10	STOP	(a_{14})		Rcl 34				Rcl 24			*
	STO 14			x<>31				x<>22			−
	STOP	(a_{21})		STO 34				x<>23		90	Rcl 23
	STO 21			XEQ Ø1				Rcl 34			*
	STOP	(a_{22})	40	Rcl ØØ				x<>32			+
	STO 22			:				x<>33			Rcl 13
	STOP	(a_{23})		STO Ø1	(x)			XEQ Ø1			Rcl 32
	STO 23			Rcl 14				Rcl ØØ			*
	STOP	(a_{24})		x<>11		70		:			Rcl 12
	STO 24			x<>12				STO Ø3	(z)		Rcl 33
20	STOP	(a_{31})		STO 14				GTO ØØ			*
	STO 31			Rcl 24				LBL Ø1			−
	STOP	(a_{32})		x<>21				Rcl 11		100	Rcl 21
	STO 32			x<>22				Rcl 33			*
	STOP	(a_{33})	50	STO 24				*			+
	STO 33			Rcl 34				Rcl 13			RTN
	STOP	(a_{34})		x<>31				Rcl 31		(104)	END

Une fois le programme appelé (XEQ α BB α) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Les valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, les inconnues x, y et z peuvent être connues en faisant Rcl Ø1 → X , Rcl Ø2 → Y et Rcl Ø3 → Z.

De nouvelles valeurs peuvent être dès lors introduites.

3/ Déterminant d'une matrice carrée [4,4] (Programme "CC")

Soit Δ de $[a_{ij}]$ pour $i,j = 1,2,3,4 \longrightarrow \Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$

Les valeurs a_{ij} devront être introduites dans les repères ij. Ex : a_{11} STO 11, ... (16 fois)

	LBL" CC		STO 41		STO 23		Rcl 42
	LBL ΦΦ	30	STOP		Rcl 33		×
	TONE 4		STO 42		x<>31		−
	STOP (a_{11})		STOP	60	STO 33		Rcl 33
	STO 11		STO 43		Rcl 43		×
	STOP (a_{12})		STOP		x<>41	90	Rcl 23
	STO 12		STO 44		STO 43		Rcl 42
	STOP (a_{13})		XEQ Φ1		XEQ Φ1		×
	STO 13		STO ΦΦ (Δ_1)		ST+ ΦΦ ($+\Delta_3$)		Rcl 22
10	STOP (a_{14})		Rcl 12		Rcl 14		Rcl 43
	STO 14		x<>11		x<>11		×
	STOP (a_{21})	40	STO 12		STO 14		−
	STO 21		Rcl 22		Rcl 24		Rcl 34
	STOP (a_{22})		x<>21	70	x<>21		×
	STO 22		STO 22		STO 24		+
	STOP		Rcl 32		Rcl 34	100	Rcl 24
	STO 23		x<>31		x<>31		Rcl 43
	STOP		STO 32		STO 34		×
	STO 24		Rcl 42		Rcl 44		Rcl 23
20	STOP		x<>41		x<>41		Rcl 44
	STO 31		STO 42		STO 44		×
	STOP	50	XEQ Φ1		ST− ΦΦ ($-\Delta_4$)		−
	STO 32		ST− ΦΦ ($-\Delta_2$)		GTO ΦΦ		Rcl 32
	STOP		Rcl 13	80	LBL Φ1		×
	STO 33		x<>11		Rcl 22		+
	STOP		STO 13		Rcl 44	110	Rcl 11
	STO 34		Rcl 23		×		×
	STOP		x<>21		Rcl 24		RTN
						(113)	END

Une fois le programme appelé (XEQ α CC α) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, le déterminant Δ peut être connu en faisant Rcl ΦΦ.

De nouvelles valeurs peuvent être dès lors introduites.

7 Système de 4 équations à 4 inconnues (x, y, z, t) (programme DD)

Soit le système suivant :

$$\left\{\begin{array}{l} a_{11}x + a_{12}y + a_{13}z + a_{14}t = a_{15} \\ a_{21}x + a_{22}y + a_{23}z + a_{24}t = a_{25} \\ a_{31}x + a_{32}y + a_{33}z + a_{34}t = a_{35} \\ a_{41}x + a_{42}y + a_{43}z + a_{44}t = a_{45} \end{array}\right\} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} a_{15} \\ a_{25} \\ a_{35} \\ a_{45} \end{bmatrix}$$

Les valeurs a_{ij} pour $i = 1,2,3,4$; $j = 1,2,3,4,5$ devront être introduites dans les repères ij. Ex: a_{11} STO 11, ... (20 fois)

```
   LBL' DD
   LBL ΦΦ
   TONE 4
   STOP   (a11)
   STO 11
   STOP   (a12)
   STO 12
   STOP   (a13)
   STO 13
10 STOP   (a14)
   STO 14
   STOP   (a15)
   STO 15
   STOP   (a21)
   STO 21
   STOP   (a22)
   STO 22
   STOP   (a23)
   STO 23
20 STOP   (a24)
   STO 24
   STOP   (a25)
   STO 25
   STOP   (a31)
   STO 31
   STOP   (a32)
   STO 32
   STOP   (a33)
   STO 33
30 STOP   (a34)
   STO 34
   STOP   (a35)
   STO 35
   STOP   (a41)
   STO 41
   STOP   (a42)
   STO 42
   STOP   (a43)
   STO 43
40 STOP   (a44)
   STO 44
   STOP   (a45)
   STO 45
   XEQ Φ1
   STO ΦΦ   (D)
   Rcl 15
   x<>11
   STO 15
   Rcl 25
50 x<>21
   STO 25
   Rcl 35
   x<>31
   STO 35
   Rcl 45
   x<>41
   STO 45
   XEQ Φ1
   Rcl ΦΦ
60 :
   STO Φ1   (x)
   Rcl 15
   x<>11
   x<>12
   STO 15
   Rcl 25
   x<>21
   x<>22
   STO 25
70 Rcl 35
   x<>31
   x<>32
   STO 35
   Rcl 45
   x<>41
   x<>42
   STO 45
   XEQ Φ1
   Rcl ΦΦ
80 :
   STO Φ2   (y)
   Rcl 15
   x<>12
   x<>13
   STO 15
   Rcl 25
   x<>22
   x<>23
   STO 25
   Rcl 35
90 x<>32
   x<>33
   STO 35
   Rcl 45
   x<>42
   x<>43
   STO 45
   XEQ Φ1
   Rcl ΦΦ
100 :
   STO Φ3   (z)
   Rcl 15
   x<>13
   x<>14
   Rcl 25
   x<>23
   x<>24
   Rcl 35
   x<>33
110 x<>34
   Rcl 45
   x<>43
   x<>44
   XEQ Φ1
   Rcl ΦΦ
   :
   STO Φ4   (t)
   STO ΦΦ
   LBL Φ1
120 XEQ Φ2
   STO Φ5   (Δi
   Rcl 12
   x<>11
   STO 12
   Rcl 22
   x<>21
   STO 22
   Rcl 32
   x<>31
130 STO 32
   Rcl 42
   x<>41
   STO 42
   XEQ Φ2
   ST- Φ5   (-l
   Rcl 13
   x<>11
   STO 13
   Rcl 23
140 x<>21
   STO 23
   Rcl 33
   x<>31
   STO 33
   Rcl 43
   x<>41
   STO 43
   XEQ Φ2
   ST+ Φ5   (+l
150 Rcl 14
   x<>11
   STO 14
   Rcl 24
   x<>21
   STO 24
   Rcl 34
```

	x<>34		Rcl 55		Rcl 24		Rcl 24		Rcl 43
	STO 34		x<>33	190	Rcl 42		*		*
	Rcl 44		x<>32		*		−		Rcl 23
160	x<>41		x<>31		−		Rcl 33		Rcl 44
	STO 44		STO 34		Rcl 33		*	210	*
	ST− 05 (−Δ4)		Rcl 44		*		Rcl 23		−
	Rcl 14		x<>43		Rcl 23		Rcl 42		Rcl 32
	x<>13	180	x<>42		Rcl 42		*		*
	x<>12		x<>41		*		Rcl 22		+
	x<>11		STO 44		Rcl 22		Rcl 43		Rcl 11
	STO 14		Rcl 05		Rcl 43	200	*		*
	Rcl 24		RTN	200	*		−		RTN
	x<>23		LBL 02		−		Rcl 34	(218)	END
170	x<>22		Rcl 22		Rcl 34		*		
	x<>21		Rcl 44		*		+		
	STO 24		*		+				

Une fois le programme appelé (XEQ α DD α) le son perçu (TONE 4) indique l'attente de l'introduction des valeurs a_{ij} (par R/S). Ces valeurs étant introduites le programme effectue son calcul. Quand celui-ci est terminé, les inconnues x, y, z et t peuvent être connues en faisant Rcl 01 → x , Rcl 02 → y , Rcl 03 → z et Rcl 04 → t.

De nouvelles valeurs peuvent être dès lors introduites.

PRGM "IF" (suite)

Je pense que vous avez déjà entendu parler de mon PRGM le "IF" et de la différence qu'est publié précédemment. Ce PRGM (Inverse Flag), comme ses initiales l'indiquent, inverse la situation initiale du DRAPeau dont le N° a été placé en X et cela suppose donc que nous connaissons au préalable, dans quel état se trouve le DRAP concerné.

```
01+LBL "XSF"
02 FC?C IND X
03 XEQ "IF"
04 STOP
05+LBL "XCF"
06 FS?C IND X
07 XEQ "IF"
08 END
```

Je vous propose donc 2 petites routines qui permettent, soit de lever, soit de baisser le DRAP. "XSF" lèvera le DRAP dans le cas ou celui-ci sera baissé et "XCF" baissera le DRAP s'il est levé.

Ceux qui n'ont pas la PPC-ROM et qui sont économes d'octets, pourront placer le PRGM ci-contre en tête du PRGM "IF" en remplaçant le LBL "IF" par LBL ØØ, en remplaçant les 2 XEQ "IF" par XEQ ØØ et en n'oubliant pas (?) d'ajouter RTN en Pas Ø8. rS (P20+T178)

Yves ALAJOUANINE
108 Bis Rue Hénon
F - 69004 LYON
Tel: 7 830 80 50

V/Réf:
n° 15
N/Réf:
YA/CD

LYON
le 29 Septembre 1983

Monsieur Philippe GUEZ
P.P.C. PARIS CHAPTER
56 rue J.-J. Rousseau
75001 PARIS

Monsieur,

Je sollicite la parution gracieuse de l'information ci-dessous dans la publication que vous dirigez.

Journée d'études sur l'emploi par les géomètres-topographes de l'ordinateur de poche HEWLETT-PACKARD HP-41CV adapté en vue de la Saisie des Observations de Topométrie Moderne, pour Logiciel de Tracer Automatique Direct (SOTM-LTAD).
Pour s'inscrire, contacter Mr Yves ALAJOUANINE tel: 7/830 80 50 108 bis rue Hénon, 69004 LYON. Possibilité de bénéficier de la convention pour la formation continue. La journée d'études est prévue chaque vendredi ou samedi (sauf jour férie), à partir du 16 Décembre 1983 sur rendez-vous par groupe réduit suivant le matériel disponible, au 108 bis rue Hénon, Lyon, de 9 a 18 h avec repas (40 Fr). Le prix modeste du stage dépend du pret de matériel au stagiaire et de la documentation conservée.

En vous remerciant vivement pour l'attention réservée à cette demande, je vous prie d'agréer Monsieur, l'expression de mes sentiments distingués.

Y Alajouanine

Yves ALAJOUANINE

CHOC EN RETOUR AU VI N° 6 P19
ORGANISATION D'UNE ROM

```
...
x00      ...ge fct

x001  10:  t=0 ...
x002  ...  t=2 utilisat.

x004  10:  2' fct
x005  01:  abc=l'instruct

x006
x007

...
...

x008
x009

x002  000  fin du
x003  000  catalogue
```

Chaque fonction du catalogue tient sur 2 mot ROM contenant:
1° mot = E00000abcd
2° mot = 00efghijkl tel que:

abcdefghjkl = adresse (3 digits). Cette adresse peut varier de 000 à FFF par page soit 4K bits
E = 0 si la fonction est en code machine
E = 1 si la fonction est en langage utilisateur

ROUTINES ET PROGRAMMES

Les adresses suivantes sont interrogéeuniquement lors:

xFF4 - D'une pause
xFF5 - Aprés chaque ligne de programme
xFF6 - Au retour d'une position d'attente
xFF7 - Lors de l'utilisation de l'interrupteur ON/OFF (fonction OFF)
xFF8 - Si le drapeau E/S est armé
xFF9 - Lors de l'utilisation de l'interrupteur ON/OFF (fonction ON)
xFFA - MEMORY LOST
Dans la plupart des cas, ces adresses contiennent ØØØ.

xFF[illegible]
xFF[illegible]
xFF[illegible] 4 adresses contenant le nom et le numero de la ROM
xFF[illegible]

xFFF contient le CHECKSUM (somme de controle) de la ROM, utilisée par la ROM diagnostique

CONTINUE AU JPC VING P20
ORGANISATION DE L'IMPRIMANTE

CHOC EN RETOUR AUX PROGRAMMES MORSE et APPRH'' V1N4

LBL EN SAUCE SINTHETIQUE

Tout d'abord, il existe deux sortes de LBL dits synthétiques.

-- les LBL ᵀalphanumériques pouvant contenir jusqu'à 14 caractères choisis parmi les 128 que reconnait l'imprimante. Que les matheux m'envoient le nombre de possibilités, ça m'intéresse car je n'ai pas réussi à le calculer. En tout cas cela laisse pas mal de possibilités, de quoi satisfaire les plus difficiles.

- les LBL ᵀsemi-locaux correspondant au complément des vrais LBL locaux (A à J et a à e). Mais ces LBL ne se retrouvent pas en mode USER comme les LBL locaux. Par contre ils peuvent être très utiles dans le cas d'un décodage rapide lettre par lettre (voir LBL "MORSE" et "APPRH").

-- les LBL alocaux pouvant recevoir des codes compris dans l'ensemble complémentaire des codes des registres internes de la 41. Ces LBL n'ont pas le petit ᵀ alphanumérique et sont invisibles lors d'un CATALOGUE.

Remarque : Dans les exemples ci-dessous, le caractére § est un code fantôme qui disparait en LBL ᵀalpha mais dont la machine se sert pour l'assignation du LBL. (voir tableau)
De même le code LBL 00 peut ne pas être le bon, suivant les machines. Dans ce cas ne pas essayer de supprimer le END généré mais faire MEMORY LOST après avoir sauvegardé vos programmes préférés mais pas celui du END.
Le code à utiliser est alors LBL 01.

LBL ᵀalpha ou semi-locaux

```
1
RCL ind 16
RCL ind 64
LBL 00 ou 01
ᵀ§A (max 15 caract )
```

remplacer par :

```
RCL 29
ᵀA (max 14 caract )   } pour GTO ᵀ

RCL 30
ᵀA (max 14 caract )   } pour XEQ ᵀ

RCL 31
ᵀA ( " 15 )           } pour W ᵀ
                        dangereuse ??
```

LBL alocaux

```
1
RCL ind 16
RCL ind 79
code sous forme
   de fonction
(RDN pour LBL M )
```

remplacer par :

```
RCL ind 80
code          } pour GTO
code

RCL ind 96
code          } pour XEQ
code
```

Je me sers surtout de la fonction W ᵀpour glisser des commentaires entre certaines lignes. Dans la plupart des cas la machine accepte bien les caractéres alphabétiques.
Mais si un jour vous n'avez rien à faire et une 41 vide de programmes intéressants, remplaçez ᵀA... par une fonction. Le W ᵀ obtenu est souvent dangereux (blockage ou Memory Lost).

LE PETIT THE'TRE DES MICROCODES

tableau a double entree
donnant la mnemonique
correspondant le mot ROM e
n hexa

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
ØØ	NOP ?NC XQ ØØ-	-ØØ ?C XQ ØØ-	A+Ø @R ?NC GO ØØ-	JNC # ØØ ?C GO ØØ-	CLRF 3 ?NC XQ Ø1-	-Ø1 ?C XQ Ø1-	A=Ø S+X ?NC GO Ø1-	JC # +ØØ ?C GO Ø1-	SETF 3 ?NC XQ Ø2-	-Ø2 ?C XQ Ø2-	A=Ø R+ ?NC GO Ø2-	JNC #+Ø1 ?C GO Ø2-	?FSET 3 ?NC XQ Ø3-	-Ø3 ?C XQ Ø3-	A=Ø ALL ?NC GO Ø3-	JC #+Ø1 ?C GO Ø3-	ØØ
Ø1	LDØR- Ø ?NC XQ Ø4-	-Ø4 ?C XQ Ø4-	A=Ø P-Q ?NC GO Ø4-	JNC #+Ø2 ?C GO Ø4-	?R = 3 ?NC XQ Ø5-	-Ø5 ?C XQ Ø5-	A=Ø XS ?NC GO Ø5-	JC # +Ø2 ?C GO Ø5-	?NC XQ Ø6-	-Ø6 ?C XQ Ø6-	A=Ø M ?NC GO Ø6-	JNC #+Ø3 ?C GO Ø6-	R = 3 ?NC XQ Ø7-	-Ø7 ?C XQ Ø7-	A=Ø MS ?NC GO Ø7-	JC #+Ø3 ?C GO Ø7-	Ø1
Ø2	XQ + GO ?NC XQ Ø8-	-Ø8 ?C XQ Ø8-	B=Ø @R ?NC GO Ø8-	JNC #+Ø4 ?C GO Ø8-	SELP 3 ?NC XQ Ø9-	-Ø9 ?C XQ Ø9-	B=Ø S+X ?NC GO Ø9-	JC # +Ø4 ?C GO Ø9-	WRIT Ø(1) ?NC XQ ØA-	-ØA ?C XQ ØA-	B=Ø R+ ?NC GO ØA-	JNC #+Ø5 ?C GO ØA-	TFI = 3 ?NC XQ ØB-	-ØB ?C XQ ØB-	B=Ø ALL ?NC GO ØB-	JC #+Ø5 ?C GO ØB-	Ø2
Ø3	?NC XQ ØC-	-ØC ?C XQ ØC-	B=Ø P-Q ?NC GO ØC-	JNC #+Ø6 ?C GO ØC-	?NC XQ ØD-	-ØD ?C XQ ØD-	B=Ø XS ?NC GO ØD-	JC # +Ø6 ?C GO ØD-	READ Ø(1) ?NC XQ ØE-	-ØE ?C XQ ØE-	B=Ø M ?NC GO ØE-	JNC #+Ø7 ?C GO ØE-	RCR 3 ?NC XQ ØF-	-ØF ?C XQ ØF-	B=Ø MS ?NC GO ØF-	JC #+Ø7 ?C GO ØF-	Ø3
Ø4	?NC XQ 1Ø-	-1Ø ?C XQ 1Ø-	C=Ø @R ?NC GO 1Ø-	JNC #+Ø8 ?C GO 1Ø-	CLRF 4 ?NC XQ 11-	-11 ?C XQ 11-	C=Ø S+X ?NC GO 11-	JC # +Ø8 ?C GO 11-	SETF 4 ?NC XQ 12-	-12 ?C XQ 12-	C=Ø R+ ?NC GO 12-	JNC #+Ø9 ?C GO 12-	?FSET 4 ?NC XQ 13-	-13 ?C XQ 13-	C=Ø ALL ?NC GO 13-	JC #+Ø9 ?C GO 13-	Ø4
Ø5	LDØR-1 ?NC XQ 14-	-14 ?C XQ 14-	C=Ø P-Q ?NC GO 14-	JNC #+ØA ?C GO 14-	?R = 4 ?NC XQ 15-	-15 ?C XQ 15-	C=Ø XS ?NC GO 15-	JC # +ØA ?C GO 15-	G=CØR;Ø ?NC XQ 16-	-16 ?C XQ 16-	C=Ø M ?NC GO 16-	JNC #+ØB ?C GO 16-	R = 4 ?NC XQ 17-	-17 ?C XQ 17-	C=Ø MS ?NC GO 17-	JC #+ØB ?C GO 17-	Ø5
Ø6	POWOFF ?NC XQ 18-	-18 ?C XQ 18-	A<>B @R ?NC GO 18-	JNC #+ØC ?C GO 18-	SELP 4 ?NC XQ 19-	-19 ?C XQ 19-	A<>B S+X ?NC GO 19-	JC # +ØC ?C GO 19-	WRIT 1 (1) ?NC XQ 1A-	-1A ?C XQ 1A-	A<>B R+ ?NC GO 1A-	JNC #+ØD ?C GO 1A-	?FI 4 ?NC XQ 1B-	-1B ?C XQ 1B-	A<>B ALL ?NC GO 1B-	JC #+ØD ?C GO 1B-	Ø6
Ø7	N=C ALL ?NC XQ 1C-	-1C ?C XQ 1C-	A<>B P-Q ?NC GO 1C-	JNC #+ØE ?C GO 1C-	?NC XQ 1D-	-1D ?C XQ 1D-	A<>B XS ?NC GO 1D-	JC # +ØE ?C GO 1D-	READ 1 (1) ?NC XQ 1E-	-1E ?C XQ 1E-	A<>B M ?NC GO 1E-	JNC #+ØF ?C GO 1E-	RCR 4 ?NC XQ 1F-	-1F ?C XQ 1F-	A<>B MS ?NC GO 1F-	JC #+ØF ?C GO 1F-	Ø7
Ø8		-2Ø ?NC XQ 2Ø-	B=A @R ?NC GO 2Ø-	JNC #+1Ø ?C GO 2Ø-	CLRF 5 ?NC XQ 21-	-21 ?C XQ 21-	B=A S+X ?NC GO 21-	JC # +1Ø ?C GO 21-	SETF 5 ?NC XQ 22-	-22 ?C XQ 22-	B=A R+ ?NC GO 22-	JNC #+11 ?C GO 22-	?FSET 5 ?NC XQ 23-	-23 ?C XQ 23-	B=A ALL ?NC GO 23-	JC #+11 ?C GO 23-	Ø8
Ø9	LDØR-2 ?NC XQ 24-	-24 ?C XQ 24-	B=A P-Q ?NC GO 24-	JNC #+12 ?C GO 24-	?R = 5 ?NC XQ 25-	-25 ?C XQ 25-	B=A XS ?NC GO 25-	JC # +12 ?C GO 25-	C=GØR;Ø ?NC XQ 26-	-26 ?C XQ 26-	B=A M ?NC GO 26-	JNC #+13 ?C GO 26-	R = 5 ?NC XQ 27-	-27 ?C XQ 27-	B=A MS ?NC GO 27-	JC #+13 ?C GO 27-	Ø9
ØA	SLC= ■ ?NC XQ 28-	-28 ?C XQ 28-	A<>C @ ■ ?NC GO 28-	JNC #+14 ?C GO 28-	SELP 5 ?NC XQ 29-	-29 ?C XQ 29-	A<>C S+X ?NC GO 29-	JC # +14 ?C GO 29-	WRIT 2(1) ?NC XQ 2A-	-2A ?C XQ 2A-	A<>C R+ ?NC GO 2A-	JNC #+15 ?C GO 2A-	?FI 5 ?NC XQ 2B-	-2B ?C XQ 2B-	A<>C ALL ?NC GO 2B-	JC #+15 ?C GO 2B-	ØA
ØB	C=N ALL ?NC XQ 2C-	-2C ?C XQ 2C-	A<>C P-Q ?NC GO 2C-	JNC #+16 ?C GO 2C-	?NC XQ 2D-	-2D ?C XQ 2D-	A<>C XS ?NC GO 2D-	JC # +16 ?C GO 2D-	READ 2(1) ?NC XQ 2E-	-2E ?C XQ 2E-	A<>C M ?NC GO 2E-	JNC #+17 ?C GO 2E-	RCR 5 ?NC XQ 2F-	-2F ?C XQ 2F-	A<>C MS ?NC GO 2F-	JC #+17 ?C GO 2F-	ØB
ØC	?NC XQ 3Ø-	-3Ø ?C XQ 3Ø-	C=B @R ?NC GO 3Ø-	JNC #+18 ?C GO 3Ø-	CLRF A ?NC XQ 31-	-31 ?C XQ 31-	C=B S+X ?NC GO 31-	JC # +18 ?C GO 31-	SETF A ?NC XQ 32-	-32 ?C XQ 32-	C=B R+ ?NC GO 32-	JNC #+19 ?C GO 32-	?FSET A ?NC XQ 33-	-33 ?C XQ 33-	C=B ALL ?NC GO 33-	JC #+19 ?C GO 33-	ØC
ØD	LDØR-3 ?NC XQ 34-	-34 ?C XQ 34-	C=B P-Q ?NC GO 34-	JNC #+1A ?C GO 34-	?R = A ?NC XQ 35-	-35 ?C XQ 35-	C=B XS ?NC GO 35-	JC # +1A ?C GO 35-	C<>GØR;Ø ?NC XQ 36-	-36 ?C XQ 36-	C=B M ?NC GO 36-	JNC #+1B ?C GO 36-	R = A ?NC XQ 37-	-37 ?C XQ 37-	C=B MS ?NC GO 37-	JC #+1B ?C GO 37-	ØD
ØE	SLCT Ø ?NC XQ 38-	-38 ?C XQ 38-	C<>B @R ?NC GO 38-	JNC #+1C ?C GO 38-	SELP A ?NC XQ 39-	-39 ?C XQ 39-	C<>B S+X ?NC GO 39-	JC # +1C ?C GO 39-	WRIT 3(1) ?NC XQ 3A-	-3A ?C XQ 3A-	C<>B R+ ?NC GO 3A-	JNC #+1D ?C GO 3A-	?FI A ?NC XQ 3B-	-3B ?C XQ 3B-	C<>B ALL ?NC GO 3B-	JC #+1D ?C GO 3B-	ØE
ØF	C<>W ALL ?NC XQ 3C-	-3C ?C XQ 3C-	C<>B P-Q ?NC GO 3C-	JNC #+1E ?C GO 3C-	?NC XQ 3D-	-3D ?C XQ 3D-	C<>B XS ?NC GO 3D-	JC # +1E ?C GO 3D-	READ 3(1) ?NC XQ 3E-	-3E ?C XQ 3E-	C<>B M ?NC GO 3E-	JNC #+1F ?C GO 3E-	RCR A ?NC XQ 3F-	-3F ?C XQ 3F-	C<>B MS ?NC GO 3F-	JC #+1F ?C GO 3F-	ØF
1Ø		-4Ø ?C XQ 4Ø-	A=C @R ?NC GO 4Ø-	JNC #+2Ø ?C GO 4Ø-	CLRF B ?NC XQ 41-	-41 ?C XQ 41-	A=C S+X ?NC GO 41-	JC # +2Ø ?C GO 41-	SETF B ?NC XQ 42-	-42 ?C XQ 42-	A=C R+ ?NC GO 42-	JNC #+21 ?C GO 42-	?FSET B ?NC XQ 43-	-43 ?C XQ 43-	A=C ALL ?NC GO 43-	JC #+21 ?C GO 43-	1Ø
11	LDØR-4 ?NC XQ 44-	-44 ?C XQ 44-	A=C P-Q ?NC GO 44-	JNC #+22 ?C ■ 44-	?R = B ?NC XQ 45-	-45 ?C XQ 45-	A=C XS ?NC GO 45-	JC # +22 ?C GO 45-	?NC XQ 46-	-46 ?C XQ 46-	A=C M ?NC GO 46-	JNC #+23 ?C GO 46-	R = B ?NC XQ 47-	-47 ?C XQ 47-	A=C MS ?NC GO 47-	JC #+23 ?C GO 47-	11
12	?P = Q ?NC XQ 48-	-48 ?C XQ 48-	A=A+B @R ?NC GO 48-	JNC #+24 ?C GO 48-	SELP B ?NC XQ 49-	-49 ?C XQ 49-	A=A+B S+X ?NC GO 49-	JC # +24 ?C GO 49-	WRIT 4(1) ?NC XQ 4A-	-4A ?C XQ 4A-	A=A+B R+ ?NC GO 4A-	JNC #+25 ?C GO 4A-	?FI B ?NC XQ 4B-	-4B ?C XQ 4B-	A=A+B ALL ?NC GO 4B-	JC #+25 ?C GO 4B-	12
13	LDI S+X ?NC XQ 4C-	-4C ?C XQ 4C-	A=A+B P-Q ?NC GO 4C-	JNC #+26 ?C GO 4C-	?NC XQ 4D-	-4D ?C XQ 4D-	A=A+B XS ?NC GO 4D-	JC # +26 ?C GO 4D-	READ 4(1) ?NC XQ 4E-	-4E ?C XQ 4E-	A=A+B M ?NC GO 4E-	JNC #+27 ?C GO 4E-	RCR B ?NC XQ 4F-	-4F ?C XQ 4F-	A=A+B MS ?NC GO 4F-	JC #+27 ?C GO 4F-	13
14		-5Ø ?C XQ 5Ø-	A=A+C @R ?NC GO 5Ø-	JNC #+28 ?C GO 5Ø-	CLRF 6 ?NC XQ 51-	-51 ?C XQ 51-	A=A+C S+X ?NC GO 51-	JC # +28 ?C GO 51-	SETF 6 ?NC XQ 52-	-52 ?C XQ 52-	A=A+C R+ ?NC GO 52-	JNC #+29 ?C GO 52-	?FSET 6 ?NC XQ 53-	-53 ?C XQ 53-	A=A+C ALL ?NC GO 53-	JC #+29 ?C GO 53-	14
15	LDØR-5 ?NC XQ 54-	-54 ?C XQ 54-	A=A+C P-Q ?NC GO 54-	JNC #+2A ?C GO 54-	?R = 6 ?NC XQ 55-	-55 ?C XQ 55-	A=A+C XS ?NC GO 55-	JC # +2A ?C GO 55-	M = C ALL ?NC XQ 56-	-56 ?C XQ 56-	A=A+C M ?NC GO 56-	JNC #+2B ?C GO 56-	R = 6 ?NC XQ 57-	-57 ?C XQ 57-	A=A+C MS ?NC GO 57-	JC #+2B ?C GO 57-	15
16	?LOWBAT ?NC XQ 58-	-58 ?C XQ 58-	A=A+1 @R ?NC GO 58-	JNC #+2C ?C GO 58-	SELP 6 ?NC XQ 59-	-59 ?C XQ 59-	A=A+1 S+X ?NC GO 59-	JC # +2C ?C GO 59-	WRIT 5(1) ?NC XQ 5A-	-5A ?C XQ 5A-	A=A+1 R+ ?NC GO 5A-	JNC #+2D ?C GO 5A-	?FI 6 ?NC XQ 5B-	-5B ?C XQ 5B-	A=A+1 ALL ?NC GO 5B-	JC #+2D ?C GO 5B-	16
17	PUSH ADR ?NC XQ 5C-	-5C ?C XQ 5C-	A=A+1 P-Q ?NC GO 5C-	JNC #+2E ?C GO 5C-	?NC XQ 5D-	-5D ?C XQ 5D-	A=A+1 XS ?NC GO 5D-	JC # +2E ?C GO 5D-	READ 5(1) ?NC XQ 5E-	-5E ?C XQ 5E-	A=A+1 M ?NC GO 5E-	JNC #+2F ?C GO 5E-	RCR 6 ?NC XQ 5F-	-5F ?C XQ 5F-	A=A+1 MS ?NC GO 5F-	JC #+2F ?C GO 5F-	17
18	?NC XQ 6Ø-	-6Ø ?C XQ 6Ø-	A=A-B @R ?NC GO 6Ø-	JNC #+3Ø ?C GO 6Ø-	CLRF B ?NC XQ 61-	-61 ?C XQ 61-	A=A-B S+X ?NC GO 61-	JC # +3Ø ?C GO 61-	SETF B ?NC XQ 62-	-62 ?C XQ 62-	A=A-B R+ ?NC GO 62-	JNC #+31 ?C GO 62-	?FSET B ?NC XQ 63-	-63 ?C XQ 63-	A=A-B ALL ?NC GO 63-	JC #+31 ?C GO 63-	18
19	LDØR-6 ?NC XQ 64-	-64 ?C XQ 64-	A=A-B P-Q ?NC GO 64-	JNC #+32 ?C G. ___	?R = B ?NC XQ 65-	-65 ?C XQ 65-	A=A-B XS ?NC GO 65-	JC # +32 ?C GO 65-	C=M ALL ?NC XQ 66-	-66 ?C XQ 66-	A=A-B M ?NC GO 66-	JNC #+33 ?C GO 66-	R = B ?NC XQ 67-	-67 ?C XQ 67-	A=A-B MS ?NC GO 67-	JC #+33 ?C GO 67-	19
1A	A=B=C=Ø ?NC XQ 68-	-68 ?C XQ 68-	A=A-1 @R ?NC GO 68-	JNC #+34 ?C GO 68-	SELP B ?NC XQ 69-	-69 ?C XQ 69-	A=A-1 S+X ?NC GO 69-	JC # +34 ?C GO 69-	WRIT 6(1) ?NC XQ 6A-	-6A ?C XQ 6A-	A=A-1 R+ ?NC GO 6A-	JNC #+35 ?C GO 6A-	?FI B ?NC XQ 6B-	-6B ?C XQ 6B-	A=A-1 ALL ?NC GO 6B-	JC #+35 ?C GO 6B-	1A
1B	POP ADR ?NC XQ 6C-	-6C ?C XQ 6C-	A=A-1 P-Q ?NC GO 6C-	JNC #+36 ?C GO 6C-	?NC XQ 6D-	-6D ?C XQ 6D-	A=A-1 XS ?NC GO 6D-	JC # +36 ?C GO 6D-	READ 6(1) ?NC XQ 6E-	-6E ?C XQ 6E-	A=A-1 M ?NC GO 6E-	JNC #+37 ?C GO 6E-	RCR B ?NC XQ 6F-	-6F ?C XQ 6F-	A=A-1 MS ?NC GO 6F-	JC #+37 ?C GO 6F-	1B
1C		-7Ø ?C XQ 7Ø-	A=A-C @R ?NC GO 7Ø-	JNC #+38 ?C GO 7Ø-	CLRF E ?NC XQ 71-	-71 ?C XQ 71-	A=A-C S+X ?NC GO 71-	JC # +38 ?C GO 71-	SETF E ?NC XQ 72-	-72 ?C XQ 72-	A=A-C R+ ?NC GO 72-	JNC #+39 ?C GO 72-	?FSET E ?NC XQ 73-	-73 ?C XQ 73-	A=A-C ALL ?NC GO 73-	JC #+39 ?C GO 73-	1C
1D	LDØR-7 ?NC XQ 74-	-74 ?C XQ 74-	A=A-C P-Q ?NC GO 74-	JNC #+3A ?C GO 74-	?R = E ?NC XQ 75-	-75 ?C XQ 75-	A=A-C XS ?NC GO 75-	JC # +3A ?C GO 75-	C<>M ALL ?NC XQ 76-	-76 ?C XQ [illegible]	A=A-C M ?NC GO 76-	JNC #+3B ?C GO 76-	R = E ?NC XQ 77-	-77 ?C XQ 77-	A=A-C MS ?NC GO 77-	JC #+3B ?C GO 77-	1D
1E	GO TO ADR ?NC XQ 78-	-78 ?C ■ 78-	C=C+C @R ?NC GO 78-	JNC #+3C ?C GO 78-	SELP E ?NC XQ 79-	-79 ?C XQ 79-	C=C+C S+X ?NC GO 79-	JC # +3C ?C GO 79-	WRIT 7(1) ?NC XQ 7A-	-7A ?C XQ 7A-	C=C+C R+ ?NC GO 7A-	JNC #+3D ?C ■ 7A-	?FI E ?NC XQ 7B-	-7B ?C XQ 7B-	C=C+C ALL ?NC GO 7B-	JC #+3D ?C GO 7B-	1E
1F	?NC XQ 7C-	-7C ?C XQ 7C-	C=C+C P-Q ?NC GO 7C-	JNC #+3E ?C GO 7C-	?NC XQ 7D-	-7D ?C ■ 7D-	C=C+C XS ?NC GO 7D-	JC # +3E ?C GO 7D-	READ 7(1) ?NC XQ 7E-	-7E ?C XQ 7E-	C=C+C M ?NC GO 7E-	JNC #+3F ?C GO 7E-	RCR E ?NC XQ 7F-	-7F ?C XQ 7F-	C=C+C MS ?NC GO 7F-	JC #+3F ?C GO 7F-	1F

	Ø	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2Ø	?NC XQ BØ-	-BØ ?C XQ BØ-	C=C+A @R ?NC [redacted]	JNC #-4Ø ?C GO BØ-	CLRF 2 ?NC XQ B1-	-B1 ?C XQ B1-	C=C+A S+X ?NC GO B1-	JC #-4Ø ?C GO B1-	SETF 2 ?NC XQ B2-	-B2 ?C XQ B2-	C=C+A R+ ?NC GO B2-	JNC #-3F ?C GO B2-	?FSET 2 ?NC XQ B3-	-B3 ?C XQ B3-	C=C+A ALL ?NC GO B3-	JC #-3F ?C GO B3-	2Ø
21	LD@R- B ?NC [redacted] B4-	-B4 ?C [redacted] B4-	C=C+A P-Q ?NC GO B4-	JNC #-3E ?C GO B4-	?R=2 ?NC XQ B5-	-B5 ?C XQ [redacted]	C=C+A XS ?NC GO B5-	JC #-3E ?C GO B5-		-B6 ?C XQ B6-	C=C+A M ?NC GO B6-	JNC #-3D ?C GO B6-	R=2 ?NC XQ B7-	-B7 ?C XQ B7-	C=C+A MS ?NC [redacted] B7-	JC #-3D ?C GO B7-	21
22	C=KEY KY ?NC XQ B8-	-B8 ?C [redacted] B8-	C=C+1 @R ?NC GO [redacted]	JNC #-3C ?C [redacted]	SELF 2 ?NC [redacted] B9-	-B9 ?C XQ B9-	C=C+1 S+X ?NC GO B9-	JC #-3C ?C GO B9-	WRIT 8(0) ?NC XQ BA-	-BA ?C XQ BA-	C=C+1 R+ ?NC [redacted]	JNC #-3B ?C [redacted] BA-	?FI 2 ?NC XQ BB-	-BB ?C XQ BB	C=C+1 ALL ?NC [redacted] BB-	JC #-3B ?C GO BB-	22
23	GO TO KEY ?NC [redacted] BC-	-BC ?C XQ BC-	C=C+1 P-Q ?NC GO BC-	JNC #-3A ?C GO BC-		-BD ?C XQ BD-	C=C+1 XS ?NC GO BD-	JC #-3A ?C GO BD-	READ 8(P) ?NC XQ BE-	-BE ?C XQ BE-	C=C+1 M ?NC GO BE-	JNC #-39 ?C GO BE-	RCR 2 ?NC XQ BF-	-BF ?C XQ BF-	C=C+1 MS ?NC GO BF-	JC #-39 ?C GO BF-	23
24	?NC XQ 9Ø-	-9Ø ?C XQ 9Ø-	C=A-C @R ?NC GO [redacted]	JNC #-38 ?C [redacted] 9Ø-	CLRF 9 ?NC XQ 91-	-91 ?C XQ 91-	C=A-C S+X ?NC GO 91-	JC #-38 ?C GO 91-	SETF 9 ?NC XQ 92-	-92 ?C XQ 92-	C=A-C R+ ?NC GO 92-	JNC #-37 ?C GO 92-	?FSET 9 ?NC XQ 93-	-93 ?C XQ 93-	C=A-C ALL ?NC GO 93-	JC #-37 ?C GO 93-	24
25	LD@R- 9 ?NC XQ 94-	-94 ?C XQ 94-	C=A-C P-Q ?NC GO 94-	JNC #-36 ?C [redacted] 94-	?R=9 ?NC XQ 95-	-95 ?C XQ 95-	C=A-C XS ?NC GO 95-	JC #-36 ?C GO 95-	T=ST ?NC XQ 96-	-96 ?C XQ 96-	C=A-C M ?NC [redacted] 96-	JNC #-35 ?C GO 96-	R=9 ?NC XQ 97-	-97 ?C XQ 97-	C=A-C MS ?NC GO 97-	JC [redacted]-35 ?C GO 97-	25
26	SET HEX ?NC XQ 98-	-98 ?C [redacted] 98-	C=C-1 @R ?NC GO 98-	JNC #-34 ?C [redacted] 98-	SELF 9 ?NC XQ 99-	-99 ?C XQ 99-	C=C-1 S+X ?NC GO 99-	JC #-34 ?C GO 99-	WRIT 9(0) ?NC XQ 9A-	-9A ?C [redacted] 9A-	C=C-1 R+ ?NC GO 9A-	JNC #-33 ?C [redacted] 9A-	?FI 9 ?NC XQ 9B-	-9B ?C XQ 9B-	C=C-1 ALL ?NC [redacted] 9B-	JC #-33 ?C GO 9B-	26
27	RAM SLCT ?NC XQ 9C-	-9C ?C XQ 9C-	C=C-1 P-Q ?NC GO 9C-	JNC #-32 ?C GO 9C-		-9D ?C XQ 9D-	C=C-1 XS ?NC GO 9D-	JC #-32 ?C GO 9D-	READ 9(0) ?NC XQ 9E-	-9E ?C XQ 9E-	C=C-1 M ?NC GO 9E-	JNC #-31 ?C GO 9E-	RCR 9 ?NC XQ 9F-	-9F ?C XQ 9F-	C=C-1 MS ?NC GO 9F-	JC #-31 ?C GO 9F-	27
28		-AØ ?C XQ AØ-	C=Ø-C @R ?NC GO AØ-	JNC #-3Ø ?C GO AØ-	CLRF 7 ?NC XQ A1-	-A1 ?C XQ A1-	C=Ø-C S+X ?NC GO A1-	JC #-3Ø ?C GO A1-	SETF 7 ?NC XQ A2-	-A2 ?C XQ A2-	C=Ø-C [redacted] ?NC GO A2-	JNC #-2F ?C GO A2-	?FSET 7 ?NC XQ A3-	-A3 ?C [redacted] A3-	C=Ø-C ALL ?NC GO A3-	JC [redacted]-2F ?C GO A3-	28
29	LD@R- A ?NC XQ A4-	-A4 ?C XQ A4-	C=Ø-C P-Q ?NC GO A4-	JNC #-2E ?C GO A4-	?R=7 ?NC XQ A5-	-A5 ?C XQ A5-	C=Ø-C XS ?NC GO A5-	JC #-2E ?C GO A5-	ST=T ?NC XQ A6-	-A6 ?C XQ A6-	C=Ø-C M ?NC GO A6-	JNC #-2D ?C GO A6-	R=7 ?NC XQ A7-	-A7 ?C [redacted] A7-	C=Ø-C MS ?NC GO A7-	JC #-2D ?C GO A7-	29
2A	SET DEC ?NC XQ A8-	-A8 ?C XQ A8-	C=C-1 @R ?NC GO A8-	JNC #-2C ?C GO A8-	SELF 7 ?NC XQ A9-	-A9 ?C XQ A9-	C=C-1 S+X ?NC GO A9-	JC #-2C ?C GO A9-	WRIT 10(0) ?NC XQ AA-	-AA ?C XQ AA-	C=C-1 R+ ?NC GO AA-	JNC #-2B ?C GO AA-	?FI 7 ?NC XQ AB-	-A[redacted] ?C XQ AB-	C=C-1 ALL ?NC GO AB-	JC #-2B ?C GO AB-	2A
2B	?NC XQ AC-	-AC ?C [redacted] AC-	C=C-1 P-Q ?NC [redacted] AC-	JNC #-2A ?C [redacted] AC-		-AD ?C XQ AD-	C=C-1 XS ?NC GO AD-	JC #-2A ?C GO AD-	READ 1[redacted] ?NC XQ AE-	-AE ?C XQ AE-	C=C-1 M ?NC [redacted] GO AE-	JNC #-29 ?C GO AE-	RCR 7 ?NC XQ AF-	-AF ?C [redacted] AF-	C=C-1 MS ?NC GO AF-	JC #-29 ?C GO AF-	2B
2C	?NC XQ B[redacted]-	-B[redacted] ?C XQ B[redacted]-	?B≠Ø @R ?NC GO BØ-	JNC #-28 ?C GO B[redacted]-	CLRF D ?NC XQ B1-	-B1 ?C [redacted] B1-	?B≠Ø S+X ?NC [redacted] B1-	JC #-28 ?C [redacted] B1-	SETF D ?NC XQ B2-	-B2 ?C XQ B2-	?B≠Ø R+ ?NC GO B2-	JNC #-27 ?C GO B2-	?FSET D ?NC XQ B3-	-B3 ?C XQ B3-	?B≠Ø ALL ?NC GO B3-	JC [redacted]-27 ?C GO B3-	2C
2D	LD@R- B ?NC XQ B4-	-B4 ?C XQ B4-	?B≠Ø P-Q ?NC GO B4-	JNC #-26 ?C GO B4-	?R=D ?NC [redacted] B5-	-B5 ?C XQ B5-	?B≠Ø XS ?NC GO B5-	JC #-26 ?C GO B5-	ST<>T ?NC XQ B6-	-B6 ?C XQ B6-	?B≠Ø M ?NC GO B6-	JNC #-25 ?C GO B6-	R=D ?NC [redacted] B7-	-B7 ?C XQ B7-	?B≠Ø MS ?NC GO B7-	JC #-25 ?C GO B7-	2D
2E	DSP OFF ?NC XQ B8-	-B8 ?C XQ B8-	?C≠Ø @R ?NC GO [redacted]	JNC #-24 ?C GO [redacted]	SELF D ?NC XQ B9-	-B9 ?C XQ B9-	?C≠Ø S+X ?NC GO B9-	JC #-24 ?C [redacted] B9-	WRIT 11(0) ?NC XQ BA-	-BA ?C XQ BA-	?C≠Ø R+ ?NC GO BA-	JNC #-23 ?C GO BA-	?FI D ?NC XQ BB-	-BB ?C XQ BB-	?C≠Ø ALL ?NC GO BB-	JC #-23 ?C GO BB-	2E
2F	WRITE DATA ?NC XQ BC-	-BC ?C XQ BC-	?C≠Ø P-Q ?NC GO BC-	JNC [redacted]-22 ?C GO BC-	?NC XQ BD-	-BD ?C XQ BD-	?C≠Ø XS ?NC GO BD-	JC #-22 ?C GO BD-	READ 11(0) ?NC XQ BE-	-BE ?C XQ BE-	?C≠Ø M ?NC [redacted] BE-	JNC #-21 ?C GO BE-	RCR D ?NC XQ BF-	-BF ?C XQ BF-	?C≠Ø MS ?NC GO BF-	JC #-21 ?C GO BF-	2F
3Ø	?NC XQ C[redacted]-	-CØ ?C XQ C[redacted]-	?ACC @R ?NC GO C[redacted]-	JNC #-2Ø ?C GO C[redacted]-	CLRF 1 ?NC XQ C1-	-C1 ?C XQ C1-	?ACC S+X ?NC GO C1-	JC #-2Ø ?C GO C1-	SETF 1 ?NC XQ C2-	-C2 ?C XQ C2-	?ACC R+ ?NC GO C2-	JNC #-1F ?C GO C2-	?FSET 1 ?NC XQ C3-	-C3 ?C XQ C3-	?ACC ALL ?NC GO C3-	JC #-1F ?C GO C3-	3Ø
31	LD@R- C ?NC XQ C4-	-C4 ?C XQ C4-	?ACC P-Q ?NC [redacted] C4-	JNC #-1E ?C GO C4-	?R=1 ?NC XQ C5-	-C5 ?C XQ C5-	?ACC XS ?NC [redacted] C5-	JC #-1E ?C GO C5-		-C6 ?C XQ C6-	?ACC M ?NC GO C6-	JNC #-1D ?C GO C6-	R=1 ?NC XQ C7-	-C7 ?C XQ C7-	?ACC MS ?NC GO C7-	JC #-1D ?C GO C7-	31
32	DSP TOG ?NC XQ C8-	-C8 ?C XQ C8-	?ACB @R ?NC GO C8-	JNC #-1C ?C GO C8-	SELF 1 ?NC XQ C9-	-C9 ?C XQ C9-	?ACB S+X ?NC GO C9-	JC [redacted]-1C ?C GO C9-	WRIT 12(0) ?NC XQ CA-	-CA ?C XQ CA-	?ACB R+ ?NC GO CA-	JNC #-1B ?C GO CA-	?FI 1 ?NC XQ CB-	-CB ?C XQ CB-	?A[redacted] ALL ?NC GO CB-	JC [redacted]-1B ?C GO CB-	32
33	FETCH S+X ?NC XQ CC-	-CC ?C XQ CC-	?ACB P-Q ?NC GO CC-	JNC #-1A ?C GO CC-		-CD ?C XQ CD-	?ACB XS ?NC GO CD-	JC #-1A ?C GO CD-	READ 12(0) ?NC XQ CE-	-CE ?C XQ CE-	?ACB M ?NC GO CE-	JNC #-19 ?C GO CE-	RCR 1 ?NC XQ CF-	-CF ?C XQ CF-	?ACB MS ?NC GO CF-	JC #-19 ?C GO CF-	33
34	?NC XQ D[redacted]-	-DØ ?C XQ DØ-	?A≠Ø @R ?NC GO DØ-	JNC #-18 ?C GO DØ-	CLRF C ?NC XQ D1-	-D1 ?C XQ D1-	?A≠Ø S+X ?NC GO D1-	JC #-18 ?C GO D1-	SETF C ?NC XQ D2-	-D[redacted] ?C [redacted] D2-	?A≠Ø [redacted] ?NC GO D2-	JNC #-17 ?C GO D2-	?FSET C ?NC XQ D3-	-D3 ?C XQ D3-	?A≠Ø ALL ?NC GO D3-	JC #-17 ?C GO D3-	34
35	LD@R- D ?NC [redacted] D4-	-D4 ?C XQ D4-	?A≠Ø P-Q ?NC GO D4-	JNC #-16 ?C GO D4-	?R=C ?NC XQ D5-	-D5 ?C XQ D5-	?A≠Ø XS ?NC GO D5-	JC #-16 ?C GO D5-	ST=C ?NC XQ D6-	-D6 ?C XQ D6-	?A≠Ø M ?NC GO D6-	JNC #-15 ?C GO D6-	R=C ?NC XQ D7-	-D7 ?C XQ D7-	?A≠Ø MS ?NC GO D7-	JC #-15 ?C GO D7-	35
36	?C RTN ?NC XQ D8-	-D8 ?C XQ D8-	?A≠C @R ?NC GO D8-	JNC #-14 ?C GO D8-	SELF C ?NC XQ D9-	-D9 ?C XQ D9-	?A≠C S+X ?NC GO D9-	JC #-14 ?C GO D9-	WRIT 13(0) ?NC XQ DA-	-DA ?C XQ DA-	?A≠C R+ ?NC GO DA-	JNC #-13 ?C GO DA-	?FI C ?NC XQ DB-	-DB ?C XQ DB-	?A≠C ALL ?NC GO DB-	JC #-13 ?C GO DB-	36
37	C=C OR A ?NC XQ DC-	-DC ?C XQ DC-	?A≠C P-Q ?NC GO DC-	JNC #-12 ?C GO DC-		-DD ?C XQ DD-	?A≠C XS ?NC GO DD-	JC #-12 ?C GO DD-	READ 13(0) ?NC XQ DE-	-DE ?C XQ DE-	?A≠C M ?NC GO DE-	JNC #-11 ?C GO [redacted]	RCR C ?NC XQ DF-	-DF ?C XQ DF-	?A≠C MS ?NC GO DF-	JC #-11 ?C GO DF-	37
38	?NC XQ E[redacted]-	-E[redacted] ?C XQ E[redacted]-	RSHFA @R ?NC GO E[redacted]-	JNC #-1Ø ?C GO E[redacted]-	CLRF Ø ?NC XQ E1-	-E1 ?C XQ E1-	RSHFA S+X ?NC GO E1-	JC #-1Ø ?C GO E1-	SETF Ø ?NC XQ E2-	-E2 ?C XQ E2-	RSHFA R+ ?NC GO E2-	JNC #-ØF ?C GO E2-	?FSET Ø ?NC XQ E3-	-E3 ?C XQ E3-	RSHFA ALL ?NC GO E3-	JC #-ØF ?C GO E3-	38
39	LD@R- E ?NC XQ E4-	-E4 ?C XQ E4-	RSHFA P-Q ?NC GO E4-	JNC #-ØE ?C GO E4-	?R=Ø ?NC XQ E5-	-E5 ?C XQ E5-	RSHFA XS ?NC GO E5-	JC #-ØE ?C GO E5-	C=ST ?NC XQ E6-	-E6 ?C XQ E6-	RSHFA M ?NC GO E6-	JNC #-ØD ?C GO E6-	R=Ø ?NC XQ E7-	-E7 ?C XQ E7-	RSHFA [redacted] ?NC GO E7-	JC #-ØD ?C GO E7-	39
3A	?NC RTN ?NC XQ E8-	-E8 ?C XQ E8-	RSHFB @R ?NC GO E8-	JNC #-ØC ?C GO E8-	SELF Ø ?NC XQ E9-	-E9 ?C XQ E9-	RSHFB S+X ?NC GO E9-	JC #-ØC ?C GO E9-	WRIT 14(0) ?NC XQ EA-	-EA ?C XQ EA-	[redacted]FB R+ ?NC GO EA-	JNC #-ØB ?C GO EA-	?FI Ø ?NC XQ EB-	-EB ?C XQ EB-	RSHFB ALL ?NC GO EB-	JC #-ØB ?C GO EB-	3A
3B	C=C AND A ?NC XQ EC-	-EC ?C XQ EC-	RSHFB P-Q ?NC GO EC-	JNC #-ØA ?C GO EC-		-ED ?C XQ ED-	RSHFB XS ?NC GO ED-	JC #-ØA ?C GO ED-	READ 14(0) ?NC XQ EE-	-EE ?C XQ EE-	RSHFB M ?NC GO EE-	JNC #-Ø9 ?C GO EE-	RCR Ø ?NC XQ EF-	-EF ?C XQ EF-	RSHFB MS ?NC GO EF-	JC #-Ø9 ?C GO EF-	3B
3C	?NC XQ F[redacted]-	-F[redacted] ?C XQ F[redacted]-	RSHFC @R ?NC GO F[redacted]-	JNC #-Ø8 ?C GO F[redacted]-	ST=Ø ?NC XQ F1-	-F1 ?C XQ F1-	RSHFC S+X ?NC GO F1-	JC #-Ø8 ?C GO F1-	CLRKEY ?NC XQ F2-	-F2 ?C XQ F2-	RSHFC R+ ?NC GO F2-	JNC #-Ø7 ?C GO F2-	?KEY ?NC XQ F3-	-F3 ?C XQ F3-	RSHFC [redacted]	JC [redacted]-Ø7 ?C GO F3-	3C
3D	LD@R- F ?NC XQ F4-	-F4 ?C XQ F4-	RSHFC P-Q ?NC GO F4-	JNC #-Ø6 ?C GO F4-	R=R-1 ?NC XQ F5-	-F5 ?C XQ F5-	RSHFC XS ?NC GO F5-	JC #-Ø6 ?C GO F5-	C<>ST ?NC XQ F6-	-F6 ?C XQ F6-	RSHFC M ?NC GO F6-	JNC #-Ø5 ?C GO F6-	R=R+1 ?NC XQ F7-	-F7 ?C XQ F7-	RSHFC MS ?NC GO F7-	JC #-Ø5 ?C GO F7-	3D
3E	RTN ?NC XQ F8-	-F8 ?C XQ F8-	LSHFA @R ?NC GO F8-	JNC #-Ø4 ?C GO F8-	SELF F ?NC XQ F9-	-F9 ?C XQ F9-	LSHFA S+X ?NC GO F9-	JC #-Ø4 ?C GO F9-	WRIT 15(0) ?NC XQ FA-	-FA ?C XQ FA-	LSHFA R+ ?NC GO FA-	JNC #-Ø3 ?C GO FA-	?FI F ?NC XQ FB-	-FB ?C XQ FB-	LSHFA ALL ?NC GO FB-	JC #-Ø3 ?C GO FB-	3E
3F	PRPH SLCT ?NC XQ FC-	-FC ?C XQ FC-	LSHFA P-Q ?NC GO FC-	JNC #-Ø2 ?C GO FC-		-FD ?C XQ FD-	LSHFA XS ?NC GO FD-	JC #-Ø2 ?C GO FD-	READ 15(0) ?NC XQ FE-	-FE ?C XQ FE-	LSHFA M ?NC GO FE-	JNC #-Ø1 ?C GO FE-	RCR F ?NC [redacted] FF-	-FF ?C XQ FF-	LSHFA MS ?NC GO FF-	JC #-Ø1 ?C GO FF-	3F

Fonctions Synthétiques et CODEs XROM

```
PRF "XS"

01♦LBL 00
02 BEEP

03♦LBL "XS
04 SF 29
05 CF 28
06 FIX 0
07 159
08 "PREF"
09 PROMPT
10 144
11 X>Y?
12 GTO 00
13 RDN
14 X>Y?
15 GTO 02
16 144
17 -
18 4
19 *
20  E
21 ÷
22 GTO "S"

23♦LBL 02
24 168
25 X>Y?
26 GTO 00
27 RDN
28 173
29 X<Y?
30 GTO 00
31 RDN
32 168
33 -
34 4
35 *
36 33
37 +

38♦LBL "S"
39 "SUFF"
40 PROMPT
41 117
42 -
43 53
44 +
45 "XROM "
46 ARCL Y
47 CF 29
48 ARCL X
49 PROMPT
50 GTO "XS"
51 END
```

Nous allons essayer de trouver les relations mathématiques qui existent entre le PREFixe et le SUFFixe d'uns FS et le CODE XRom correspondant.

La TABLE des CODES nous indique le PREFixe et le SUFFixe de chaque FS.

1°) PREFIXE

Un premier groupe de PREF compris entre 144 à 159 correspond aux CODEs XR Ø1 à 61. Nous constatons sur la liste ci-après que pour un INCrément de 1 sur le PREF CODE F, l'INC de l'XR est de 4 et nous pourrons donc exploiter cette relation "élémentaire".

Un second groupe de PREF est compris entre 168 et 174 mais nous allons exclure 174 que nous verrons plus loin. Nous retrouvons le même INC de 4 mais nous constatons que les CODEs XR 33 à 53 sont déjà utilisés pour les PREF 152 à 157.

Je ne sais pas, hélas, comment notre Diva s'y retrouve (?) et je lance un S.O.S. à ce sujet.

Le PREF 174 est particulièrement "merdique" car il donne différents CODEs XR suivant le SUFF utilisé. Je lance également un appel à ce sujet et j'indique simplement que les XR correspondants peuvent être 158, 174 & 2Ø6 et le PREF 175 est également opérant.

2°) SUFFIXE

Le problème est plus simple car les SUFF sont tous compris de 117 à 127 et les Amateurs d'HEX vont être satisfaits.....

Nous constatons que le premier DIGit du SUFF/CF est toujours F et que le 1er DIG des XR est toujours 3. De plus, les seconds DIG sont toujours égaux. Il semblait alors simple d'exploiter ces relations pour faire les convertions mais je me suis bien vite aperçu que l'utilisation de l'arithmétique élémentaire était bien plus simple.

Vous trouverez le PRGM "XS" (Xrom Synth) qui effectue ces convertions. Une "barrière" interdit de rentrer des PREF non compatibles mais aucune protection n'a été prévue pour les SUFF.

L'utilisation de ce PRGM ne pose pas de problèmes particuliers (E peut être remplacé par 1).

J'attends donc un exposé détaillé sur les CODEs XR des FS. MERCI d'avance et le collègue dévoué peut être assuré de ma profonde et sincère reconnaissance.

				DEC	HEX	DEC	HEX
144	"donne"	XR	Ø1	53	35	117	75
145	d°		Ø5	54	36	118	76
146	d°		Ø9	55	37	119	77
147	d°		13	56	38	12Ø	78
148	d°		17	57	39	121	79
149	d°		21	58	3A	122	7A
15Ø	d°		25	59	3B	123	7B
151	d°		29	6Ø	3C	124	7C
152 & 168	"donnent"	XR	33,	61	3D	125	7D
153 & 169	d°		37,	62	3E	126	7E
154 & 17Ø	d°		41,	63	3F	127	7F
155 & 171	d°		45,				
156 & 172	d°		49,				
157 & 173	d°		53,				

rS (P20)

"CDX" et "DCX"

```
01♦LBL "CDX"
02 1-
03 APOT
04 ATOX
05 FS? 00
06 XTOA
07 STOP
08♦LBL "DCX"
09 XTOA
10 END
```

rS

```
CF 00
FIX 0
ABCDE
XEQ "CDX"
69   ***
ABCD
RUN
ABCDE
CLX
SF 00
XEQ "CDX"
69   ***
ABCDE
47
XEQ "DCX"
ABCDE
```

* (129 octets)

Ceux qui possèdent la PPC-ROM connaissent les PRGM "CD" & "DC"* et n'auront pas l'utilité des 2 routines ci-jointes.qui sont donc destinées aux "autres" qui disposeront cependant du module XFUNCTIONS (vendu actuellement 275NF).

Le PRGM "CDX" (Codes de Droite en X, en français) place en X le CODE DEC du CAR de Droite d'une chaîne placée en ALPHA et agit donc comme ATOX mais sur le CAR de droite et si vous levez le DRAP ØØ, le CAR ne sera pas effacé (en fait, il est effacé par ATOX mais réintroduit par XTOA).

Le PRGM "DCX" est tout simplement la fonction XTOA qui ajoute à la Droite de ALPHA, le CAR dont le CODE a été placé (ou se trouve) en X.

Ceux qui ont lu mon "papier" sur le module EXTENDED I/O se souviendront (?) que la fonction ATOXR fait la même chose que "CDX" en effaçant systématiquement le CAR de Droite. La fonction ATOXX permet de ne pas effacer ce CAR en la faisant précéder de: ALENG 1 - mais ceux qui disposent de ce module auront trouvé la combine....

En conclusion, ceux qui ont la chance d'avoir la PPC-ROM, ont en permanence à leur disposition, une quantité appréciable de PRGM bien utiles et qui n'utilisent aucun octet en MEM mais les "déshérités" peuvent se consoler en utilisant les merveilleuses fonctions du XF.

Il est évident que "CDX" & "DCX" ne s'adressent, n'ont été écrits qu'à l'attention des débutants.

Amitiés à tous, rS (P20)

P.S.: Le Pas Ø2 peut être remplacé par -1 et 1- a pour CODES DEC 17,28 et peut également être obtenu avec la FS "-" ayant les CODES DEC d'ASN 1 et 26 et je vous donnerai ultérieurement les CODES des FS "spécial'course" que j'utilise régulièrement et qui permettent de "TOUT FAIRE" sans utiliser des PRGM du genre "LB" et "KA" et ceci, avec une rapidité surprenante (pour un "sportif" entraîné).....

S.O.S.

Qui a trouvé ou trouvera le moyen de sauvegarder sur carte(s) magnétique(s) l'ensembles des alarmes y compris les XYZALM.

R/S

J.P. SANDOZ PPC R18

DRÔLES DE FONCTIONS !

Le format d'affichage(FIX,ENG ou SCI et le nombre de chiffres affichés après la virgule) est contrôlé par 6 bits du registre d:les flags 36 à 4I.
Les drapeaux 40 et 4I codent le mode d'affichage de la façon suivante:

Drapeau: 40	4I	
0	0:	SCI
0	I:	ENG
I	0:	FIX
I	I:	FIX/ENG

Le nombre de chiffres après la virgule est codé par les 4 drapeaux 36 à 39:

Drapeau:	36	37	38	39
Correspondance binaire:	8	4	2	I

Ainsi,si l'on est en FIX 9,le format d'affichage sera codé :

36	37	38	39	40	4I
I	0	0	I	I	0
8	+		I	FIX	
	=9				

Une première constatation est que 4 bits(I digit) permettent de compter jusqu'à I5 et non 9.Toutefois un FIX I5 synthétique,bien qu'il ait les drapeaux 36 à 39,se comportera comme FIX 0.

Une remarque plus intéressante est que les trois fonctions:ENG,FIX et SCI,peuvent,par l'intermédiaire de la PS,admettre des arguments compris entre 0 et I27!!!

En résumé,la HP-4I doit normalement compter de 0 à 9,mais elle peut en fait compter jusqu'à I5,et on pourra,avec un peu de PS,lui demander de compter de 0 à I27!!!Donc,si on fait FIX e,qui correspond à la fonction synthétique FIX(I27),la HP doit résoudre un problème très amusant:compter jusqu'à I27 sans dépasser I5.....

L'expérience montre en fait que l'exécution d'un FIX e intervient sur les modes DEG,RAD et GRAD, et mes recherches m'ont permis de comprendre le mécanisme de manipulation des drapeaux dans le cas d'arguments supérieurs à I5.Tout d'abord,un coup d'oeil au bouquin de Jean Daniel DODIN nous montre que les modes DEG,RAD et GRAD sont codés dans les drapeaux 42 et 43 comme suit:

Drapeau: 42	43	
0	0	DEG
0	I	RAD
I	0	GRAD
I	I	RAD

INTERPRETATION:L'argument maximum que l'on peut obtenir est I27,qui s'écrit en binaire:
I27(dec)=IIIIIII(Bin).
Lors de l'exécution d'une fonction contrôlant le mode d'affichage avec un argument supérieur à I5,la HP-4I commence par recopier les 4 bits de droite de l'argument(en binaire) avec les drapeaux 39 à 36...

Jusque là,tout se déroule normalement.Mais où la HP va-t'elle caser les autres bits?? Il reste en effet I à 3 bits supplémentaires,selon la grandeur de notre argument artificiel!!
Eh bien notre machine préférée ne trouve rien d'autre à faire que de continuer à coder l'argument sur les drapeaux 43 à 4I!
Seulement,la chose n'est pas si simple qu'on le pense car le comportement de notre machine est pour le moins "bizarre":
.Si l'argument est supérieur à I5,la HP va essayer de coder le cinquième bit (en partant de la droite) sur le drapeau 43.
    Si elle doit le désarmer:elle n'y touche PAS!!!
    Si elle doit l'armer,deux cas se présentent:
        a)le flag était à 0,elle l'arme et continue le codage s'il y a plus de 5 bits à coder (argument supérieur à 3I).
        b)le flag était à I,elle le désarme et reporte l'opération sur le flag situé à gauche! (ici le flag 42).
Ainsi,la réaction en chaîne se poursuit jusqu'à ce que la HP trouve un flag à 0.J'allai oublier de dire que lorsqu'on éxécute par exemple FIX e,la HP-4I commence par enclencher le mode FIX avec les flags 40 et 4I avant de coder l'argument(postfixe).Une fois le 5ième bit codé,le codage du sixième débute à partir du drapeau 42 et se poursuit de la même façon.Enfin,le septième bit est codé dans les mêmes conditions à partir du flag 4I.

Ce mécanisme étant finalement assez compliqué,je me propose de vous le montrer sur des exemples:

<u>Ier exemple:</u> départ FIX 4,mode DEG.
exécution de FIX e.

36	37	38	39	40	4I	42	43	
0	I	0	0	I	0	0	0	Flags au départ.
I	I	I	I	I	0	0	0	Passage en FIX et codage des 4 bits du début.
I	I	I	I	I	I	I	I	Les trois autres bits sont codés sans problème car les flags 4I à 43 étaient à 0.

<u>2em exemple:</u>départ ENG 4,mode RAD.
exécution de FIX e.

36	37	38	39	40	4I	42	43	
0	I	0	0	0	I	0	I	Flags au départ.
I	I	I	I	I	0	0	I	Passage en FIX et codage des 4 bits.

Pour coder le cinquième bit de I27=IIIIIII(en partant de la droite),la HP va au flag 43 qu'elle veut armer.Comme il l'est déjà,elle le désarme et arme le 42 qui ne l'était pas.On a donc:

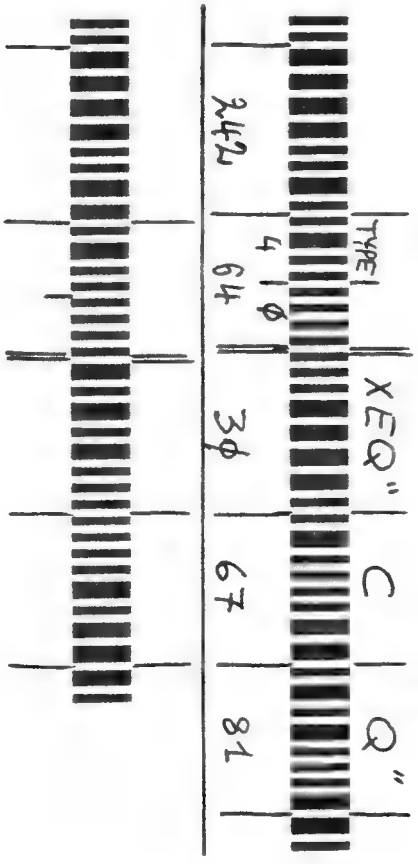
I I I I I 0 I 0

Le codage du sixième bit commence alors. La HP veut armer le flag 42 qui l'est déjà: elle le désarme et arme le 4I(qui était à O).
Le codage du septième bit commence ensuite au flag 4I.La HP va désarmer tous les flags en se déplaçant vers la gauche jusqu'à ce qu'elle en trouve un qui soit désarmé!Cette réaction en chaîne se poursuit au delà du flag 36,et pourrait même aller jusqu'au drapeau O si on armait les drapeaux O à 4I.
Toutefois celà semble difficile à réaliser,puisque le drapeau 30 est presque toujours désarmé.C'est sans doute faisable avec de la PS,et on pourrait observer ce qui se produit lorsqu'on atteint le drapeau O(...?...).
Ce processus trouvera peut-être une application dans certains programmes,mais il est indispensable de bien le maîtriser...Je vous laisse donc faire "joujou" avec ces drôles de fonctions.....

HP

J.BOSCHAT(P5I+T7I)

PS:je viens de vérifier ce qui ce produit lorsque notre "réaction en chaîne" arrive jusqu'au drapeau O:elle continue en reprenant au drapeau 55!



CB pour stockage de données par séquences

Vous m'excuserez pour cette traduction "pas très claire" (Sequenced DATA Bar CODE) mais, rassurez-vous, vous allez voir que c'est plus simple que cela ne parait.

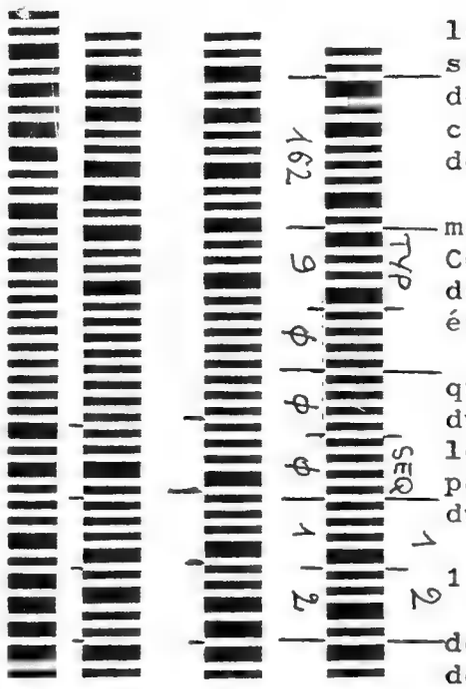
Les CB correspondants servent en fait, à être utilisés avec les Fonctions WNDDTX & WNDDTA du lecteur de CB.

Si vous incorporez dans un PRGM :
1Ø,Ø19 WNDDTX l'affichage W: SC TO 1Ø vous demandera de balayer le RANG de CB contenant la VAL NUM qui sera STOckée dans le REG 1Ø puis demandera le 11 et ainsi de suite jusqu'à 19 et stockera les VAL lues successivement dans les REG correspondants et en vérifiant à chaque fois si le RANG lu est le RANG convenable.

Si vous remplacez WNDDTX par WNDDTA, ce seront des données ALPHA qui seront stockées dans les REG et les chaînes devront donc avoir un nbre de CAR inférieur ou égal à 6.

Il se peut que vous ne voyez pas bien l'utilité de ces 2 F mais je vous assure que, pour des applications professionnelles, c'est beaucoup plus agréable qu'avec les F RDTAX ou RDTA du lecteur de cartes.

Ces CB sont du type 9 (NUM) ou 1Ø (ALPHA) et leur représentation est la suivante :
Le 1er O contient la somme de contrôle de la même façon que pour les CB du type 4 vus précédemment.
Le 3ème ½ O contient le type de CB (9 ou 1Ø).
Les trois ½ O suivants contiennent le N° de la séquence et c'est cette VAL qui contrôle le bon ordre d'entrée des RANGs successifs.



Si nous reprenons l'exemple précité (1Ø,Ø19), le 1er RANG contiendra le N° de séquence 1Ø et, seul ce RANG sera accepté lors de la première demande. Donc, si par exemple vous balayez le RANG contenant le N° 15 alors que c'est le 16 qui est demandé, vous aurez le message W: SEQ ERR.

Sur 12 BITs nous pouvons compter jusqu'à 4095 mais le manuel précise que la VAL maximale est 999. Cela me semble "farfelu" compte tenu du nbre de REG dont nous disposons et je pense qu'un Octet aurait été suffisant (256 REG).

Les ½ O suivants sont utilisés de la même façon que pour les CB du type 6 pour les données NUM et du type 7 pour les données ALPHA. Pour les VAL NUM, le nbre minimal d'O CB est donc de 3 et ne cherchez pas les CB du type 7; ils feront l'objet avec ceux du type 8 (ALPHA APPEND), d'un autre "papier".

Nous verrons également les fonctions codées sur 1 Octet qui sont assez particulières.

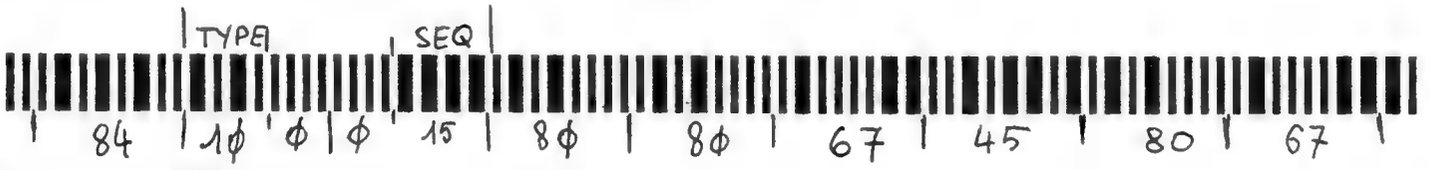
Si cela vous intéresse, je pourrais vous parler des possibilités offertes par la F WNDSCN du lecteur de CB qui offre des possibilités qui me semblent dignes d'intérêt. Si je n'ai pas encore réussi à aller "AU FOND DE LA HP-41", je vais essayer d'aller au fond des CB .....

```
LIST 004

18♦LBL "WX"
19 5 E-3
20 WNDDTX
21 END
```

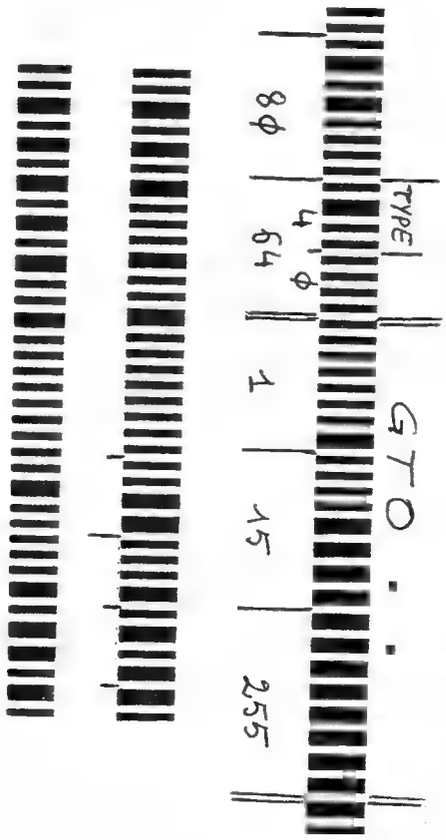
J'espère que vous avez réussi à suivre, sans trop de difficultés, mes indications. Je tiens cependant à vous assurer que c'est plus facile à comprendre (même lorsque je lis (?) l"américain"!) qu'à écrire.

Pardonnez mes maladresses; je ne suis qu'un Amateur, Amateur passionné il est vrai mais cela ne suffit pas toujours pour se faire comprendre....

Bonnes PRGMmations, rS (P20)



CB à exécution directe



Ce type de CB (4) permet l'exécution directe, c'est-à-dire aussitôt après lecture, de fonctions codées sur plus de 2 Octets de CB (ne pas confendre avec les F codées sur 2 O qui, si je ne me trompe pas, ne sont pas toutes codables sur 2 O de CB), comme GTO .. XEQ "CQ" (exemples).

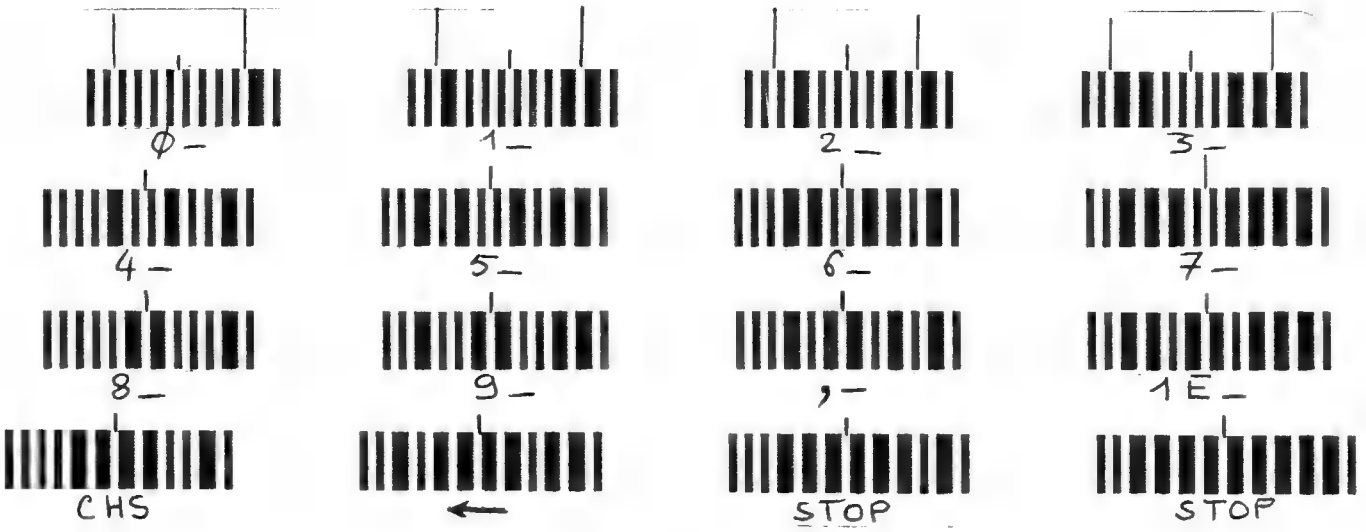
Le 1er O contient la somme de contrôle qui est sigma VAL DEC des O suivants, MOD 255 de la même façon que pour les PRGM/CB (types 1 & 2).

Le 3ème ½ O contient le type de CB (4) et, comme le ½ O suivant est Ø, le 2ème O est donc 4Ø (HEX) ce qui donne 64 en DEC (les "fanas" de l'HEX vont être contents....).

Les O suivants sont les "Octets-porteurs" et en consultant les exemples ci-joints, en particulier le premier RANG de CB et en vous aidant de votre bréviaire, je pense que vous devriez piger.

Les VAL DEC utilisées pour GTO .. sont issues du tableau de la Page 219 du manuel PLOTTER.

Indiquons que ce type de CB est utilisable en mode RUN ainsi qu'en mode PRGM mais dans ce dernier cas, il est évident que les instructions ne sont pas exécutées mais écrites car nous savons (?) que l'exécution automatique d'un PRGM est commandée par le DRAPeau 11 que nous pouvons lever en imprimant un RANG de CB correspondant; du type 4, bien entendu....



1 Octet de CB

Seul le 2ème ½ O est "utilisé" (de poid fort ou "porteur") et nous en déduisons donc, qu'il n'est possible d'y introduire que les entiers positifs de Ø à 15.

Pour le 1er ½ O, les Américains indiquent qu'il est "Mirror Image Checksum" ce qui pourrait se traduire : Somme de Contrôle à "image miroir" (?).

Je préfèrerais parler de symétrie mais l'examen des CB ci-joints vous permettra de choisir le terme qui vous convient....

Vous constaterez que les possibilités de codage sur 1 octet de CB sont limitées. Un genre de miroir aux alouettes, somme toute !....

Mise à part la représentation de la somme de contrôle, ce genre de CB qui est du type 5, n'est pas aussi compliqué qu'il parait..... rS (P20)

Fonction WALL

Cette F du lecteur de CARTes [redacted] parait intéressante et j'utilise beaucoup sa soeur WRTA, du lecteur de K7. L'ennui est que cette F WALL [redacted] été bâclée par les concepteurs de chez HP et utilise donc un nbre de CARTes fixe (1) quelque soit le nbre de REG utilisés dans la MEM principale. On [redacted] comprend pas (ou on comprend mal ?) pourquoi, [redacted] pour l'enregistrement de données (WDTA [redacted] WDTAX), le nbre de CARTes nécessaire n'est pas proportionnel au nbre de données à sauvegarder.

Si [redacted] par exemple, un PRGM de 50 REG qui utilise 100 REG dans lesquels sont STOckées des données "prédéfinies" et 19 autres REG de données, l'ensemble occupe 169 REG et ne devrait donc nécessiter que 11 PISTes (169/16) au MAXIMUM car il est évident que ces REG viendraient s'ajouter à ceux utilisés pour STOcker les autres informations.

Je pense que seul le MC peut permettre de résoudre ce problème et je [redacted] comprends pas pourquoi personne n'a réalisé cette F "miracle"(ou ce PRGM "miracle" ?). La raison de cette carence est peut-être due [redacted] fait que le lecteur de K7 résoud (?) [redacted] problème tout en adoptant le même procédé [redacted] 336 REG utilisés quelque soit la "place" réellement utilisée. Signalons que, compte tenu de la capacité d'une K7, ce gâchis est plus facilement supportable mais n'en reste pas moins inadmissible de la part des concepteurs de chez HP qui devraient être des professionnels "méticuleux".....

Tous ceux qui pourraient avoir des renseignements complémentaires sur ces 2 F (WALL [redacted] WRTA) sont invités à nous les communiquer. MERCI !

(1) Cherchez donc dans le manuel HP82104A (P27), combien faut-il de CARTes pour ne pas risquer le MEMORY LOST annoncé à la P28 ! La solution peut être trouvée en divisant les 336 REG précités, par 16 (nbre de REG STOckable sur une PISTe) et ne prenez-pas votre machine pour faire l'opération, cela nous donne 21 PISTes ou 11 CARTes.... mais à ce prix, vous avez tout de même les ASN qui sont enregistrées.....

Au plaisir de lire ce que vous avez découvert,
rS (P20+T178)

Petites découvertes

Ceux qui ont la chance de posséder la PPC-ROM utiliseront les CODES 162 & 128 (avec "MK") pour ASN l'XROM 1Ø,ØØ à une touche et découvriront l'affichage de

C PPC 1981

en mode PRGM. Ceux qui savent vont sourire car les CODES XROM de la ROM ont comme préfixe 1Ø et vous pourrez donc en déduire que, par exemple, le préfixe du XF étant 25, XROM 25,ØØ nous donne les CODES DEC 166,64 (utiliser "CDXR") mais ASN 166,64 nous répond ROM en mode PRGM et

-EXT FCN 1B

en mode RUN. Vous allez [redacted] dire :

<< A quoi ça sert ? >>

Je vous pose la même question....

Nous finirons peut-être un jour par aller "AU FOND de la 41C" ? ? ? ?

J'indique à ceux qui préfèrent le CRIC ("restons Français !) que les CODES permettant de l'ASN sont 241,67 et non pas 161, 67 comme mon PRGM "CDXR" le laisserait supposer.Il ne nous reste plus qu'à trouver la formule mathématique nous permettant d'obtenir les CODES ASN correspondant aux fonctions genre CRIC, BG, eGOBEEP, ↑-, μ-, QLOADER, etc... Au plaisir de lire vos (?) réponses. Allez, la FRANCE !

Je crois utile d'indiquer que c'est le DRAPeau Ø3 qui est armé lorsqu'il n'y a plus de papier dans l'IMP (OUT OF PAPER) car je n'ai pas lu cela dans les manuels de la "41" (?) et je pense que cela peut être utile dans certain cas particuliers. Winfried MASCHKE le savait puisque dans son merveilleux PRGM "BAP" (dont "P" est issu), il a utilisé cette possibilité pour "faire repartir" le PRGM (Pas 217 FS? Ø3 Pas 218 GTO 14 etc...).

Philippe, tu devrais demander à Richard NELSON qu'il t'envoie des Numéros "SPECIAL ISSUE D 1983" du PPC Journal que tu pourrais envoyer aux adhérents intéressés par une adhésion à notre CLUB "Père" (et "frère"?) des USA (anglophobes, s'abstenir!). Ce N° contient entre autres, le fameux PRGM "BAP" précité et constitue, à mon avis, une très bonne publicité que nous leur devons bien, n'est-ce-pas ? rS (P20)

Fonctions Synthétiques en CB/PRGM

```
01♦LBL "FS"
02 SF 25
03 PINIT
04 SF 21
05 CF 29
06 FIX 8
07 ADV
08 XEQ "ABC"
09 "S"
10 ASTO X
11 1,05
12 BCP
13 R↑
14 CHS
15 BCO
16 END
```

PRGM ABC → voir JPC PRECEDENT

Je vous assure que créer les CB correspondant aux FS n'est pas facile et j'ai longuement et vainement cherché à créer ces CB en TYPes 4 ou 5.

J'ai tout de même réussi à créer les CB des FS mais uniquement en mode PRGM et vous trouverez ci-contre le PRGM que j'utilise. Il doit être accompagné du PRGM "ABC" que j'ai envoyé à notre Président et nécessite le module PLOTTER.....

La lecture des CB (voir exemple ci-joint) est simple mais nécessite, lorsque vous lisez : W: RDY Ø2 de taper BSP (←) puis : PRGM (ON) vous permet de lire la Fonction "rentrée". Cette petite contrainte m'a permis de n'utiliser que 5 octets alors que mon collègue et homonyme Jacob G. SCHWARTZ de Philadelphie en utilise 8.

Je reste donc à votre disposition pour vous fournir les CB des FS non imprimées par notre collègue "US".

Utilisation du PRGM "FS"

Placer en ALPHA le nom de la FS, exemples : RCL M (SPACE admis), STO O, RCL IND a, etc...

XEQ "FS" et c'est parti !

J'oubliais le principal : il est indispensable de placer au préalable, en MEM principale, la Fonction Synthétique suivie de LBL "S".

JPC va peut-être devenir le spécialiste des CB.

rS (P20)

X < > d

r S

PLANCHES DE CB DES INSTRUCTIONS PORTANT SUR LES REGISTRES D'ETAT:

Ces codes-barre se présentent tous sous la forme de programmes d'une ligne(C'est sous cette seule forme que la HP-41 accepte ces instructions) , et ils exigent certaines précautions d'emploi.Il y a deux méthodes pour utiliser les trois planches de codes-barre:

A)Balayer tous les codes portant sur les registres d'Etat,puis insérer les autres instructions;

1 Introduire le label global du programme.
2 Balayer le code de END.
3 Balayer le code-barre de la première instruction désirée sur la planche de CB.
4 Balayer le code de END.
5 Balayer le code-barre de l'instruction suivante désirée sur la planche.
6 Balayer le code de END.

⋮

7 Balayer la dernière instruction synthétique.
8 En mode PRGM,effectuez [GTO] [.] [ALPHA] (label global introduit en 1) [ALPHA]
9 Insérez toutes les autres instructions à partir du clavier,en supprimant les END provisoires.

B)Balayer les codes portant sur les registres d'Etat au fur et à mesure de l'introduction du prgm:

1 Introduisez le label global du programme,puis toutes les instructions jusqu'à la première instruction portant sur les registres d'Etat.
2 Balayer le code de END.
3 Balayer le code-barre de l'instruction synthétique désirée.
4 En mode PRGM,effectuez [GTO] [.] [ALPHA] (Label global du programme) [ALPHA]
5 Balayer les codes de [BST] [←] et [SST]
6 L'instruction est en place.Continuez à introduire le programme jusqu'à l'instruction synthétique suivante et retournez en 2.

JAKE SCHWARTZ
PPC n° 1820.

La traduction de l'anglais est de JEAN BOSCHAT

(1)

END PACK

	STO	RCL		STO IND	RCL IND
M			M		
N			N		
O			O		
P			P		
Q			Q		
T			T		
a			a		
b			b		
c			c		
d			d		
e			e		
x<> M		view	ISG M		
N			N		
O			O		
P			P		
Q			Q		
T			T		
a			a		
b			b		
c			c		
d			d		

②

END

SCAN LINE

GTO - (ALPHA / LBL)

DST

←

SST

	ASTO	ARCL		ASTO IND	ARCL IND	
M			M			M
N			N			N
O			O			O
P			P			P
Q			Q			Q
T			T			T
a			a			a
b			b			b
c			c			c
d			d			d
e			e			e

x<>(in) M		VIEW IND	ISE IND		DSE IND	M
N			N			N
O			O			O
P			P			P
Q			Q			Q
T			T			T
a			a			a
b			b			b
c			c			c
d			d			d
e			e			e

③

END | SCAN LINE | GTO · (ALPHA LBL) | BST | ← | SST

	ST +	ST −		ST *	ST /	
M			M			M
N			N			N
O			O			O
P			P			P
Q			Q			Q
r			r			r
a			a			a
b			b			b
c			c			c
d			d			d
e			e			e
M			M			M
N			N			N
O			O			O
P			P			P
Q			Q			Q
r			r			r
a			a			a
b			b			b
c			c			c
d			d			d
e			e			e

X<Y?	X>Y?	X<=Y?	Σ+
Σ-	HMS+	HMS-	MOD
%	% CH	P-R	R-P
LN	X↑2	SQRT	Y↑X
CHS	E↑X	LOG	10↑X
E↑X -1	SIN	COS	TAN
ASIN	ACOS	ATAN	DEC
1/X	ABS	FACT	X≠Ø?
X>Ø?	LN1+X	X<Ø?	X=Ø?
INT	FRC	D-R	R-D
HMS	HR	RND	OCT
CLΣ	X<>Y	PI	CLST
R↑	RDN	LASTX	CLX
X=Y?	X≠Y?	SIGN	X<=Ø?
MEAN	SDEV	AVIEW	CLD
DEG	RAD	GRAD	ENTER↑
STOP	RTN	BEEP	CLA
ASHF	PSE	CLRG	AOFF
AON		PROMPT	ADV
RCL--	STO--	ST+--	ST- --
ST*--	ST/--	ISG--	DSE--
VIEW--	ΣREG--	ASTO--	ARCL--
FIX-	SCI-	ENG-	TONE-
↑--	μ--	ØBEEP-	XROM 12,00
S-	XROM 20,00	XROM 24,00	eGØBEEP--
SF--	CF--	FS?C--	FC?C--

FS? --	FC? --	GTO IND ØØ
SG -	STO ØØ	λ --
GTO Ø3	GTO Ø4	GTO Ø5
GTO Ø7	-	e ≠ x̄ --
2 --	GTO 12	GTO 13
END	END	END
DL a x̄ --		
2 --		
--	a --	X<> --
--	GTO --	B --
GTO Ø2	GTO ØØ	d< --
Σ+ --	A -	GTO ØØ
Ø ≤ 9 --	GTO ØØ	GTO ØØ
Ø --	XEQ a --	-
END	DC A BCS - a --	ST + NIA --
	XEQ ØØ	ST+NIA --
< --	XEQ ØØ	X EQ ØØ
LBL --	GTO ØØ	GTO ØØ
9 --	GTO ØØ	GTO ØØ
GTO ØØ	ACLP --	2 --
GTO ØØ	Σ+ --	GTO ØØ
GTO ØØ	GTO ØØ	> --
GTO ØØ	2 --	A -
--	XEQ ØØ	> --
X EQ ØØ	ST+NIA --	GTO ØØ
XEQ ØØ	ST+NIA --	GTO ØØ
X EQ ØØ	x̄ --	2 --

XROM 8,0

XROM 8,1

XROM 8,2

XROM 8,3

XROM 8,4

XROM 8,5

XROM 8,6

XROM 8,7

XROM 8,8

XROM 8,9

XROM 8,10

XROM 8,11

XROM 8,12

XROM 8,13

XROM 8,14

XROM 8,15

XROM 8,16

XROM 8,17

XROM 8,18

XROM 8,19

XROM 8,20

XROM 8,21

XROM 8,22

XROM 8,23

XROM 8,24

XROM 8,25

XROM 8,26

XROM 8,27

XROM 8,28

XROM 8,29

XROM 8,30

XROM 8,31

XROM 8,32

XROM 8,33

XROM 8,34

XROM 8,35

XROM 8,36

XROM 8,37

XROM 8,38

XROM 8,39

XROM 8,40

XROM 8,41

XROM 8,42

XROM 8,43

XROM 8,44

XROM 8,45

XROM 8,46

XROM 8,47

XROM 8,48

XROM 8,49

XROM 8,50

XROM 8,51

XROM 8,52

XROM 8,53

XROM 8,54

XROM 8,55

XROM 8,56

XROM 8,57

XROM 8,58

XROM 8,59

XROM 8,60

XROM 8,61

XROM 8,62

XROM 8,63

XROM 9,0
XROM 9,1
XROM 9,2
XROM 9,3
XROM 9,4
XROM 9,5
XROM 9,6
XROM 9,7
XROM 9,8
XROM 9,9
XROM 9,10
XROM 9,11
XROM 9,12
XROM 9,13
XROM 9,14
XROM 9,15
XROM 9,16
XROM 9,17
XROM 9,18
XROM 9,19
XROM 9,20
XROM 9,21
XROM 9,22
XROM 9,23
XROM 9,24
XROM 9,25
XROM 9,26
XROM 9,27
XROM 9,28
XROM 9,29
XROM 9,30
XROM 9,31
XROM 9,32
XROM 9,33
XROM 9,34
XROM 9,35
XROM 9,36
XROM 9,37
XROM 9,38
XROM 9,39
XROM 9,40
XROM 9,41
XROM 9,42
XROM 9,43
XROM 9,44
XROM 9,45
XROM 9,46
XROM 9,47
XROM 9,48
XROM 9,49
XROM 9,50
XROM 9,51
XROM 9,52
XROM 9,53
XROM 9,54
XROM 9,55
XROM 9,56
XROM 9,57
XROM 9,58
XROM 9,59
XROM 9,60
XROM 9,61
XROM 9,62
XROM 9,63

XROM 10,0
XROM 10,1
XROM 10,2
XROM 10,3
XROM 10,4
XROM 10,5
XROM 10,6
XROM 10,7
XROM 10,8
XROM 10,9
XROM 10,10
XROM 10,11
XROM 10,12
XROM 10,13
XROM 10,14
XROM 10,15
XROM 10,16
XROM 10,17
XROM 10,18
XROM 10,19
XROM 10,20
XROM 10,21
XROM 10,22
XROM 10,23
XROM 10,24
XROM 10,25
XROM 10,26
XROM 10,27
XROM 10,28
XROM 10,29
XROM 10,30
XROM 10,31
XROM 10,32
XROM 10,33
XROM 10,34
XROM 10,35
XROM 10,36
XROM 10,37
XROM 10,38
XROM 10,39
XROM 10,40
XROM 10,41
XROM 10,42
XROM 10,43
XROM 10,44
XROM 10,45
XROM 10,46
XROM 10,47
XROM 10,48
XROM 10,49
XROM 10,50
XROM 10,51
XROM 10,52
XROM 10,53
XROM 10,54
XROM 10,55
XROM 10,56
XROM 10,57
XROM 10,58
XROM 10,59
XROM 10,60
XROM 10,61
XROM 10,62
XROM 10,63

XROM 11,0
XROM 11,1
XROM 11,2
XROM 11,3
XROM 11,4
XROM 11,5
XROM 11,6
XROM 11,7
XROM 11,8
XROM 11,9
XROM 11,10
XROM 11,11
XROM 11,12
XROM 11,13
XROM 11,14
XROM 11,15
XROM 11,16
XROM 11,17
XROM 11,18
XROM 11,19
XROM 11,20
XROM 11,21
XROM 11,22
XROM 11,23
XROM 11,24
XROM 11,25
XROM 11,26
XROM 11,27
XROM 11,28
XROM 11,29
XROM 11,30
XROM 11,31
XROM 11,32
XROM 11,33
XROM 11,34
XROM 11,35
XROM 11,36
XROM 11,37
XROM 11,38
XROM 11,39
XROM 11,40
XROM 11,41
XROM 11,42
XROM 11,43
XROM 11,44
XROM 11,45
XROM 11,46
XROM 11,47
XROM 11,48
XROM 11,49
XROM 11,50
XROM 11,51
XROM 11,52
XROM 11,53
XROM 11,54
XROM 11,55
XROM 11,56
XROM 11,57
XROM 11,58
XROM 11,59
XROM 11,60
XROM 11,61
XROM 11,62
XROM 11,63

Index des fonctions

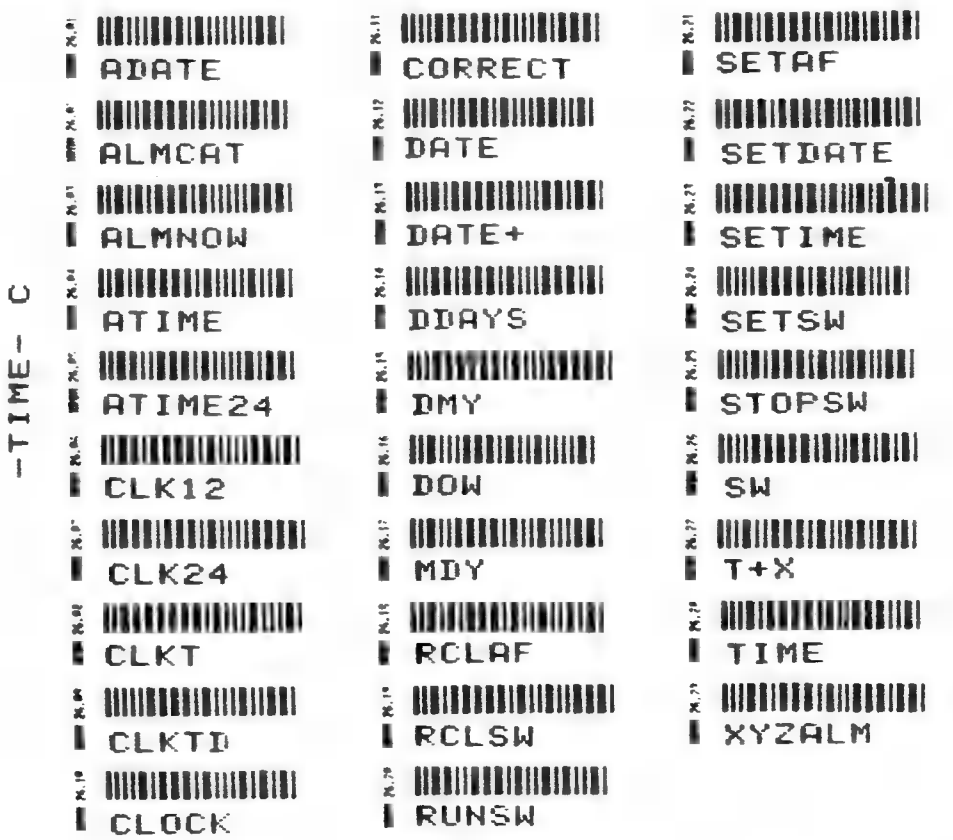
Fonction	Description	Registre ALPHA	Registre X
ALENG 25,01 (page 15)	Place dans X la longueur du contenu de ALPHA.		
ANUM 25,02 (page 15)	Balaye ALPHA à la recherche d'un nombre et en place la valeur dans le registre X.		
APPCHR 25,03 (page 30)	Ajoute le contenu de ALPHA comme nouvel enregistrement à la fin de l'enregistrement en cours dans le fichier travail en mémoire d'extension.	Chaîne	
APPREC 25,04 (page 29)	Ajoute le contenu de ALPHA comme nouvel enregistrement à la fin du fichier travail en mémoire d'extension.	Chaîne	
ARCLREC 25,05 (page 32)	Ajoute un enregistrement ou une partie d'enregistrement du fichier travail au contenu de ALPHA. La copie commence à la position instantanée du pointeur et se poursuit jusqu'à ce que le registre ALPHA soit plein ou que l'enregistrement soit épuisé.		
AROT 25,06 (page 16)	Effectue une permutation circulaire du contenu de ALPHA du nombre de caractères spécifié par X (vers la gauche si ce nombre est positif et vers la droite s'il est négatif).		Nombre de caractères et sens de la permutation.
ATOX 25,07 (page 14)	Supprime le caractère le plus à gauche de ALPHA, le convertit en un code numérique et place ce code dans X.		
CLFL 25,08 (page 24)	Efface un fichier données ou ASCII de la mémoire d'extension.	Nom de fichier.	
CLKEYS 25,09 (page 13)	Efface toutes les affectations de touches.		
CRFLAS 25,10 (page 24)	Crée un fichier ASCII en mémoire d'extension.	Nom de fichier.	Nombre de registres.
CRFLD 25,11 (page 24)	Crée un fichier de données en mémoire d'extension.	Nom de fichier.	Nombre de registres.
DELCHR 25,12 (page 30)	Supprime du fichier travail le nombre de caractères spécifié par X en commençant à la position instantanée du pointeur.		Nombre de caractères à supprimer.
DELREC 25,13 (page 29)	Supprime du fichier travail l'enregistrement indiqué par le pointeur.		
EMDIR 25,14 (page 23)	Liste les fichiers présents en mémoire d'extension.		
FLSIZE 25,15 (page 25)	Place dans X le nombre de registres du fichier nommé.	Nom de fichier.	
GETAS 25,16 (page 33)	Copie le fichier nommé de l'unité de stockage de masse dans la mémoire d'extension.	Fichier de stockage de masse, fichier en mémoire d'extension.	

PURFL 25,31 **(page 24)**	Supprime le fichier nommé de la mémoire d'extension.	Nom de fichier.	
RCLFLAG 25,32 **(page 11)**	Rappelle dans X les données représentant l'état des indicateurs 00 à 43.		
RCLPT 25,33 **(page 27)**	Place dans X un nombre représentant les positions des pointeurs dans le fichier travail (*rrr* pour les fichiers données et *eee,ccc* pour les fichiers ASCII) ou y place le nombre d'octets du fichier s'il contient un programme.		
RCLPTA 25,34 **(page 27)**	Place dans X un nombre représentant les positions des pointeurs (ou le nombre d'octets) dans le fichier spécifié par le contenu de ALPHA (*rrr* pour un fichier de données, *eee,ccc* pour un fichier ASCII et *ooo* pour un fichier programme).	Nom de fichier.	
REGMOVE 25,35 **(page 11)**	Copie *nnn* registres de la mémoire principale, commençant au registre *ooo*, à un nouvel endroit commençant au registre *ddd*.		*ooo,dddnnn.*
REGSWAP 25,36 **(page 11)**	Echange *nnn* registres commençant au registre *ooo* avec un bloc de même taille commençant au registre *ddd*.		*ooo,dddnnn.*
SAVEAS 25,37 **(page 33)**	Copie le fichier ASCII spécifié de la mémoire d'extension dans le fichier spécifié de la mémoire de masse.	Fichier en mémoire d'extension, fichier en mémoire de masse.	
SAVEP 25,38 **(page 25)**	Copie le programme nommé de la mémoire principale dans le fichier programme nommé de la mémoire d'extension.	Nom de programme, nom de fichier.	
SAVER 25,39 **(page 27)**	Copie tous les registres de la mémoire principale dans le fichier de données nommé.	Nom de fichier.	
SAVERX 25,40 **(page 27)**	Copie dans le fichier de données travail, le bloc de registres de la mémoire principale spécifié par le contenu de X.		*ddd,fff* (début et fin du bloc de registres à copier).
SAVEX 25,41 **(page 28)**	Copie le contenu de X dans le fichier de données travail à la position courante du pointeur.		
SEEKPT 25,42 **(Seite 26)**	Positionne les pointeurs du fichier travail conformément aux indications données dans X.		*rrr* (fichier de données). *rrr,ccc* (fichier ASCII).
SEEKPTA 25,43 **(page 26)**	Positionne les pointeurs du fichier spécifié dans ALPHA conformément aux indications données dans X.	Nom de fichier.	*rrr* (fichier de données). *rrr,ccc* (fichier ASCII).
SIZE? 25,44 **(page 17)**	Place dans X le nombre de registres de données de la mémoire principale.		
STOFLAG 25,45 **(page 11)**	Remet les indicateurs binaires 00 à 43 (ou une partie d'entre eux) dans un état précédent. Utilise les données obtenues en exécutant RCLFLAG.		X: Etats des indicateurs ou *dd,ff* (début et fin du bloc d'indicateurs). Y: Etats des indicateurs.
X<>F 25,46 **(page 12)**	Echange le contenu de X avec l'état des indicateurs 0 à 7.		Code d'état (0–255).
XTOA 25,47 **(page 14)**	Convertit le contenu de X en son caractère équivalent et ajoute ce dernier à ALPHA.		Code de caractère (0–255).

Fonction	Description	Registre ALPHA	Registre X
GETKEY 25,17 (page 16)	Arrête l'exécution du programme jusqu'à ce qu'une touche soit enfoncée ou qu'environ 10 secondes se soient écoulées. Place dans X le code de la touche ou 0 si aucune touche n'a été utilisée.		
GETP 25,18 (page 25)	Remplace le dernier programme en mémoire principale par le contenu du fichier programme nommé.	Nom de fichier programme.	
GETR 25,19 (page 28)	Copie le fichier de données nommé en mémoire principale en commençant au registre 00.	Nom de fichier de données.	
GETREC 25,20 (page 32)	Efface ALPHA et y copie l'enregistrement du fichier travail identifié par la position instantanée du pointeur d'enregistrement. Les caractères sont copiés jusqu'à ce que la fin de l'enregistrement ou un maximum de 24 caractères soit atteint.		
GETRX 25,21 (page 28)	Copie tous les registres (ou une partie) du fichier travail dans les registres spécifiés de la mémoire principale. La copie commence à la position instantanée des pointeurs dans le fichier travail.		*ddd.fff* (début et fin du bloc de registres de la mémoire principale dans lequel les données doivent être transférées).
GETSUB 25,22 (page 25)	Copie le programme nommé de la mémoire d'extension à la fin de l'espace programme de la mémoire principale.	Nom de fichier programme.	
GETX 25,23 (page 28)	Copie le contenu du registre courant du fichier travail dans X.		
INSCHR 25,24 (page 31)	Insère le contenu de ALPHA devant la position instantanée des pointeurs dans le fichier travail ASCII.	Caractères à insérer.	
INSREC 25,25 (page 29)	Insère le contenu de ALPHA comme nouvel enregistrement devant l'enregistrement courant du fichier travail ASCII.	Caractères à insérer.	
PASN 25,26 (page 13)	Affecte par programme une fonction ou un programme à une touche du clavier.	Nom de fonction ou de programme.	Code de touche.
PCLPS 25,27 (page 17)	Supprime de la mémoire principale le programme nommé et tous les suivants.	Nom de programme.	
POSA 25,28 (page 15)	Balaye ALPHA à la recherche du ou des caractères contenus dans X et place dans X la position du premier caractère de la chaîne ou −1 si la chaîne n'est pas trouvée.		Chaîne ALPHA ou code de caractère.
POSFL 25,29 (page 31)	Balaye le fichier travail ASCII à la recherche d'une chaîne identique au contenu de ALPHA. Place dans X la position enregistrement-caractère du premier caractère de la chaîne ou −1 si la chaîne n'est pas trouvée.	Chaîne.	
PSIZE 25,30 (page 17)	Définit par programme la répartition de la mémoire principale entre registres de données et programme (fonction SIZE programmable).		Nombre de registres de données à allouer.

Index des fonctions

Numéro XROM			
26.01	ADATE	Ajoute un nombre représentant une date au registre ALPHA	Page 20
26.02	ALMCAT	Fournit un listage des alarmes; lorsqu'elle est arrêtée, réaffecte les touches du clavier à des fonctions non programmables du Catalogue des alarmes.	Page 43
26.03	ALMNOW	Active les alarmes de commande périmées.	Page 52
26.04	ATIME	Ajoute un nombre représentant une heure au registre ALPHA.	Page 19
26.05	ATIME24	Ajoute un nombre au format 24 heures au registre ALPHA.	Page 20
26.06	CLK12	Fait passer l'affichage du calculateur au format 12 heures.	Page 14
26.07	CLK24	Fait passer l'affichage du calculateur au format 24 heures.	Page 14
26.08	CLKT	Fait passer le calculateur en mode affichage de l'heure seulement.	Page 16
26.09	CLKTD	Fait passer le calculateur en mode affichage de l'heure et de la date.	Page 16
26.10	CLOCK	Affiche l'horloge.	Page 16
26.11	CORRECT	Met l'horloge à l'heure et règle le facteur d'exactitude.	Page 58
26.12	DATE	Place le nombre représentant la date actuelle dans le registre X.	Page 18
26.13	DATE+	Calcule une nouvelle date à partir d'une date connue et du nombre de jours.	Page 23
26.14	DDAYS	Calcule le nombre de jours entre deux dates.	Page 24
26.15	DMY	Commute le format de la date sur jour-mois-année.	Page 13
26.18	RCLAF	Rappelle le facteur d'exactitude de l'horloge.	Page 58
26.19	RCLSW	Place l'heure du chronomètre dans le registre X.	Page 37
26.20	RUNSW	Déclenche le chronomètre.	Page 37
26.21	SETAF	Spécifie le facteur d'exactitude de l'horloge.	Page 59
26.22	SETDATE	Met l'horloge à une date donnée.	Page 14
26.23	SETIME	Met l'horloge à une heure donnée.	Page 14
26.24	SETSW	Met le chronomètre sur l'heure de départ spécifiée.	Page 36
26.25	STOPSW	Arrête le fonctionnement du chronomètre.	Page 37
26.26	SW	Met le calculateur en mode chronomètre et réaffecte les touches du clavier aux fonctions non programmables du chronomètre.	Page 29
26.27	T+X	Règle l'heure de l'horloge à un facteur spécifié.	Page 17
26.28	TIME	Place le nombre représentant l'heure actuelle dans le registre X.	Page 18
26.29	XYZALM	Enclenche l'alarme.	Page 41
■	ON	Affiche l'horloge. Non programmable	Page 16
	DOW	26.16	
	MDY	26.17	



ACA
ADV
ABS
ALPHA
AOFF
AON
ASTO
ARCL
ASHF
AVIEW
BST
CLA
DSE
FACT
FRC
INT
FC?
FC?C
FS?C
E↑X-1
LN1+X
MOD
MEAN
PACK
PROMPT
PSE
R↑
RND
SDEV
SIGN
SIZE
STOP
ST+
ST-
ST*
ST/
TONE
X<Y?
X≠Y?
X<>..
X≠0?
X<0?
X<=0?
X>0?

DEC
DEG
D-R
GRAD
HR
HMS
HMS+
HMS-
OCT
RAD
R-D
%CH
CLD
CLP
CLRG
CLST
DEL
BST
SST
APPEND
COPY
SIZE
<..
END

LBL A ASN a/

GTO.000
GTO.001

GTO END
CLP*

TISSERAND
42, Av. du Gal de Gaulle
77500 CHELLES

```
PRF "FLAG
*LBL "FLAG"
02 64
03 STO 01
04 AON
FONCTION ?"
06 AVIEW
07 STOP
08 ASTO 10
09 AOFF
10 "CODE ?"
11 AVIEW
12 STOP
13 STO 02
 "NUMERO ?"
15 AVIEW
16 STOP
17 STO 03

18*LBL 15
19 CLA
20 ARCL 10
21 ARCL 03
22 ACA
23 3
24 BCREGX
25 1
26 +
27 BCCKSK
28 CHS
29 BCO
30 ADV
31 ADV
32 ADV
33 ADV
34 ADV
35 ADV
36 ADV
37 1
38 ST+ 03
39 GTO 15
40 .END.
```

SF00
SF01
SF02
SF03
SF04
SF05
SF06
SF07
FS?00
FS?01
FS?02
FS?03
FS?04
FS?05
FS?06
FS?07
CF00
CF01
CF02
CF03
CF04
CF05
CF06
CF07
SF12
CF28

FS?C 00
FS?C 01
FS?C 02
FS?C 03
FS?C 04
FS?C 05
FS?C 06
FS?C 07
FC?00
FC?01
FC?02
FC?03
FC?04
FC?05
FC?06
FC?07
FC?C 00
FC?C 01
FC?C 02
FC?C 03
FC?C 04
FC?C 05
FC?C 06
FC?C 07
FS?C 26
CF29

TISSERAND
42, Av. du Gal de G
77500 CHELLES

E =1

E1=10

E2=100

E3=1.000

E4=10.000

E5=100.000

E6=1.000.000

E7=10.000.000

E8=100.000.000

E9=1.000.000.000

E-1=0,1

E-2=0,01

E-3=0,001

E-4=0,0001

E-5=0,00001

E-6=0,000001

E-7=0,0000001

E-8=0,00000001

E-9=0,000000001

-E=-1

-E1=-10

-E2=-100

-E3=-1.000

-E4=-10.000

-E5=-100.000

-E6=-1.000.000

-E7=-10.000.000

-E8=-100.000.000

-E9=-1.000.000.000

FIX 0

FIX 1

FIX 2

FIX 3

FIX 4

FIX 5

FIX 6

FIX 7

FIX 8

FIX 9

TISSERAND
42, Av. du Gal de Gaulle
77500 CHELLES

PPC PARIS CHAPTER

Association regie par la loi de 1901, enregistree
a Paris le 2 decembre 1982 sous le numero 82/3240

Bulletin d'adhesion

Nom :_:_:_: